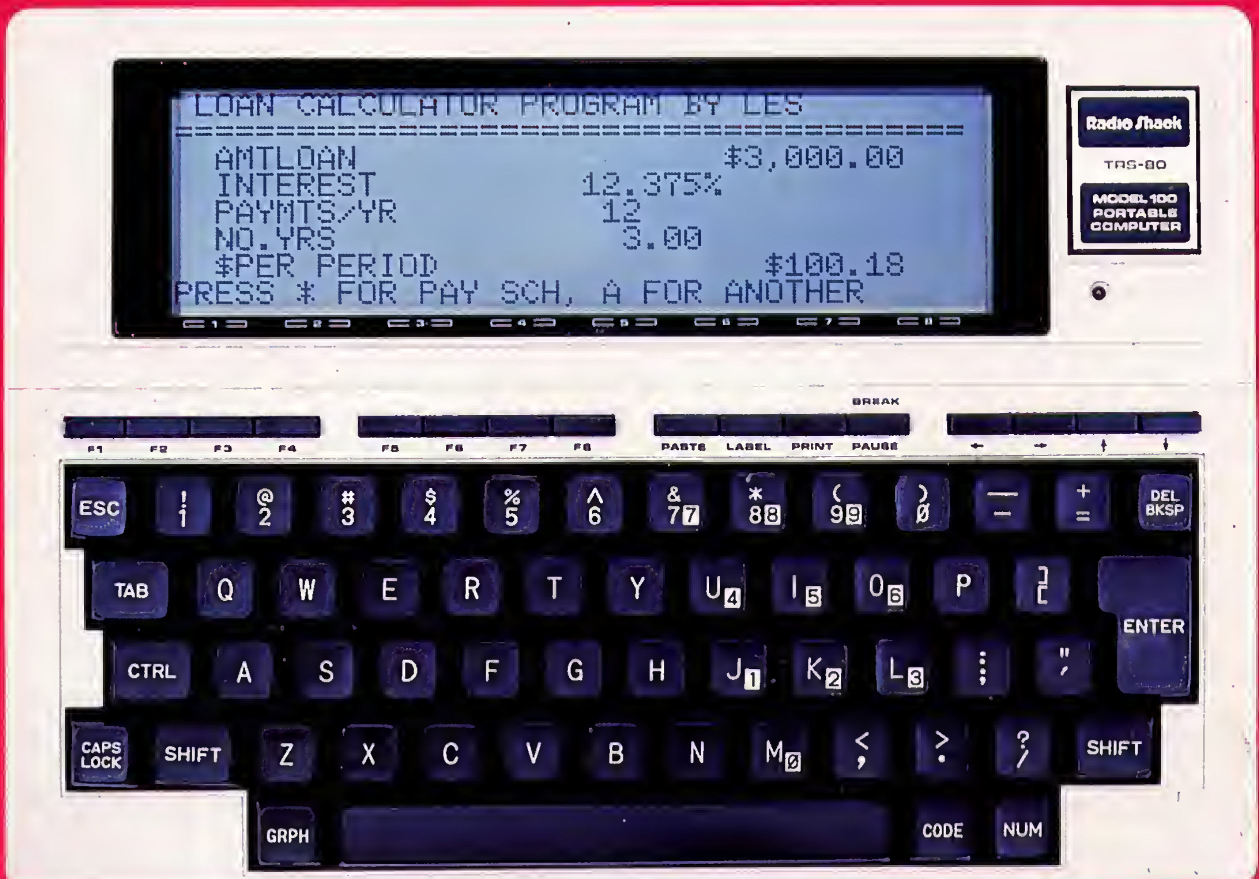
1799

FINANCIAL DECISION MAKING WITH YOUR TRS-80™ MODEL 100

INCLUDING 18 PROGRAMS



```
LOAN CALCULATOR PROGRAM BY LES
========================================
  AMTLOAN                     $3,000.00
  INTEREST          12.375%
  PAYMTS/YR          12
  NO.YRS              3.00
  $PER PERIOD                   $100.18
PRESS * FOR PAY SCH, A FOR ANOTHER
```

BY LESLIE SPARKS

FINANCIAL DECISION MAKING WITH YOUR
TRS-80™ MODEL 100
INCLUDING 18 PROGRAMS

BY LESLIE SPARKS

TAB TAB BOOKS Inc.
BLUE RIDGE SUMMIT, PA. 17214

Also by the Author from TAB Books Inc.

No. 1479 *Investment Analysis with Your Microcomputer*

FIRST EDITION
FIRST PRINTING

Library of Congress Cataloging in Publication Data

Sparks, L.E.
Financial decision making with your TRS-80 model 100,
including 18 programs.

Includes index.
1. Finance, Personal—Data processing. 2. Business
enterprises—Finance—Data processing. 3. Finance—Data
processing. 4. TRS-80 Model 100 (Computer)—Programming.
I. Title.
HG179.S5528 1984 332.024'0028'54 84-8704
ISBN 0-8306-0799-4
ISBN 0-8306-1799-X (pbk.)

Contents

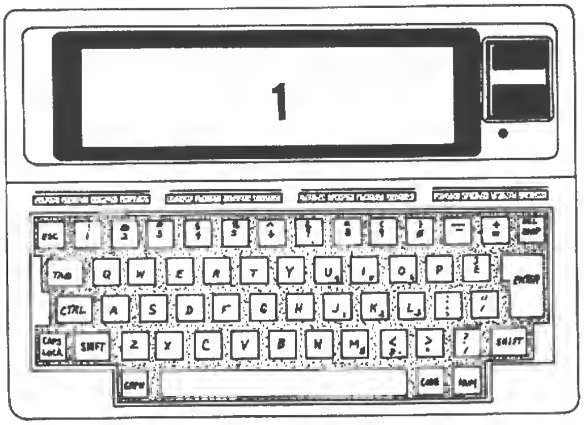
Introduction

This book can turn your TRS-80 Model 100 computer into a powerful assistant for financial decision making. There are two kinds of tools in this book. First, there are the programs. These are designed to provide you with an easy way to carry out the messy calculations necessary to make better financial decisions. The programs make the computer work on the details, leaving you free to work on the understanding.

The second tool is the text. The text is the documentation for the programs. It will tell you how to run the programs—what numbers to put where and which buttons to push. It will do more than that, too. The text is designed to help you better understand what the calculations mean and why the various calculations are important.

The text is not a complete course on financial decisions and financial calculations. It would take more than one book to provide that type of coverage, but the text does provide sufficient coverage of the topics to allow you to make informed use of the programs. A short bibliography has been provided for those of you who want or need additional information. I hope that you find I was successful in providing enough information to help you use the programs.

Although the programs are written specifically for the TRS-80 Model 100, they can run with modification on a wide variety of other computers as well. Appendix A discusses converting the programs to run on CP/M systems using Microsoft's MBASIC.



The TRS-80™ Model 100

The purpose of this chapter is to emphasize some of the features of the Model 100 that either are not documented or are particularly useful for the task at hand with this book. This section is not a replacement for the user's manual that comes with the Model 100. You should refer to it when you have questions.

The first thing to remember about the Model 100 is that *everything is in memory.* This is important when you are editing a program. The only copy of the program is the one you are working on. Any changes you make in the program automatically become permanent. Thus, if you blunder, there is no way to undo the blunder and go back to the program the way it was before you erred. This can cause serious problems and long-term loss of sleep, good humor, and hair.

Therefore I strongly recommend that you have backups of the program available just in case. Now, in the case of the Model 100, you can have your backups as tape files or as RAM files. I recommend that you use both forms of backup. The tape backup will provide you with a way to start over without retyping the entire program if everything goes bad. The RAM backup provides a convenient way of protecting from blunders and is much quicker than tape.

Making a tape backup is simple. You enter BASIC from the Model 100's master menu by placing the cursor over the word BASIC and then pressing ENTER. The BASIC sign-on message will be displayed and you're ready to go. Load the program you want to work with by typing LOAD“MYPROGR. Then press ENTER. When you see OK on the screen, you're ready to save the program to tape.

First make sure that the recorder is ready. Make sure that the various connectors are plugged in according to the instructions in the user's manual. Then make sure that the tape (and not the leader) is under the recording head of the recorder. Simultaneously press the RECORD and PLAY buttons on the recorder and then type CSAVE“MYPROGR and then press ENTER. The recorder should turn on and the program will be saved. When the program is saved and the recorder stops,check to make sure that the program was saved without error. First rewind the tape to the start of your program. Then type CLOAD"MYPROGR"? and press ENTER. This is the load-and-verify command. The program on tape will be compared with the program in memory. If the two programs are not the same, the computer will display VERIFY FAILED. If the verify worked, you'll see OK after the recorder stops. If you get the message VERIFY FAILED, first try reloading the program with a slightly different volume setting on the recorder. If that fails, try resaving the program and then verifying the second attempt.

If you continue to get VERIFY FAILED messages, try cleaning the recorder. Radio Shack and other companies sell special cassettes that clean and demagnetize the heads of a cassette recorder. You really do need one.

Making a RAM file backup copy of your program requires you to "trick" the computer. To see what I mean, type the following program into your Model 100 and then save it to RAM with the command SAVE"MYPRO.BA". Now try to resave it with the command SAVE"BACKUP.BA". You will get a ?FC error message. The computer will not allow you to resave the program as a BASIC tokenized file.

So how do you make your backup? Save the program as an ASCII file. Try this to save MYPROG—type SAVE"MYPROG.DO",A. No error. You have saved it as a *document* (.DO) in non-tokenized form. This ASCII file then becomes your backup. As soon as you have the program debugged and successfully saved on tape,you should kill the ASCII backup file and free the memory.

When you start debugging programs, you'll soon learn that all program editing is done with the text editor contained in the Model 100's ROM. This takes some getting used to—especially if you're used to the BASIC editor that is used by most other versions of Microsoft BASIC. The big advantage of the editor is that it allows you to correct only those parts of a particular line of code that are wrong. Some simple computers don't have editors and require that you retype the entire line of code to correct an error.

Note that when you enter the editor you are in *insert* mode. The material you type will be inserted into the line. The Model 100 editor does not use the edit commands that other Microsoft BASIC editors use.

You select the editor by typing EDIT NN, where NN is the line number you want to edit. (If you want to work on the entire program type EDIT. This will load all of the program into the text editor.) The screen is cleared and the line you want to edit is displayed at the top of the display. You can change any part of the line that you want—even the line number. And you have all the features of the text editor available to help you make the changes.

The ability to change line numbers with the editor can be a lifesaver. BASIC requires that every line have a number and that the number must be an integer. This can be a real problem when you're writing a program and then discover that you need to fit additional lines of code after Line 9 but before Line 10. If you have this problem with the Model 100, just type EDIT 10 and, when the display shows the line, change the 10 to 15 (or whatever) and then exit EDIT by pressing function key F8. Now you have the space you need to fit in your new lines of code.

You can also change the line number to a number totally out of sequence. For example, assume that you have a program with Lines 10, 20, and 30. You discover that Line 20 should follow Line 30, not precede it. You can fix the problem by retyping the entire line as Line 40. Or you can EDIT 20, change the line number from 20 to 40, and then return to BASIC. The BASIC interpreter will rearrange the order of the lines to correspond to the sequence of line numbers.

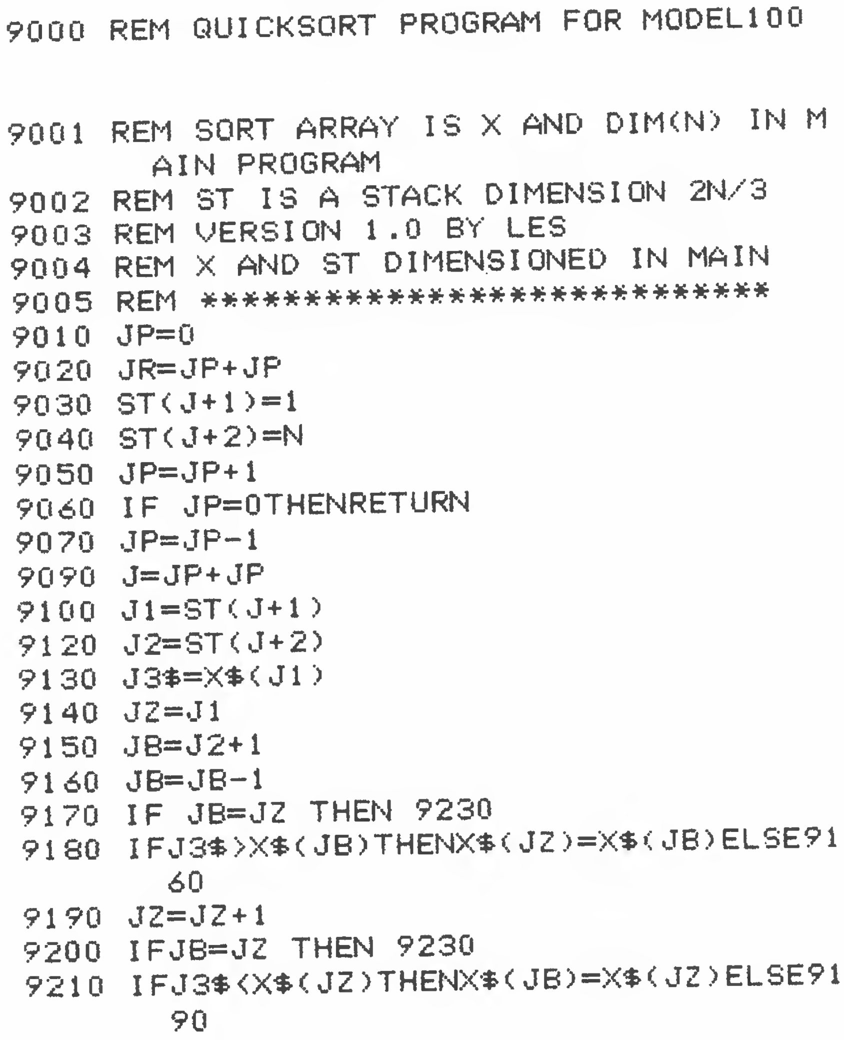
When you need to make changes in several lines, another useful trick is to load as many lines into the edit buffer as you want. For example, assume that you need to make changes in Lines 100, 110, 120, 130, 140, 150, 160, 170, 180, 190, and 200. You could edit each line separately, a somewhat tedious task. Or you can issue the command EDIT 100-200. This command will load all the desired lines into the editor for editing.

One situation where the ability to edit a large block of code is useful is the case where you have several programs that use a block of more-or-less identical code. An example is the data entry format (Chapter Two) used for most of the programs in this book. Much of the code is the same for all the programs, but there are differences, too. Instead of typing each block over and over, I created an ASCII file of the block and then merged it into the program I was working on. I then used the ability to edit a large block of code to make the changes necessary to fit the basic block into the program.

I found that all the editing commands were useful. For example the *find string:* command is useful to locate where changes need to be made. The *cut* command is used to delete long blocks of unneeded code, and the *paste* command is used to move blocks of code around. And the copy command can be used to duplicate often-used lines of code. I suggest that you use these ideas in your own programming. The time saving can be quite large.

Let's create a small BASIC program just to demonstrate the techniques. The program will get data from the keyboard, sort the data, and then display it on the display. The sort program is likely to be quite useful, so we will create it as a separate program that can be reused. The sort program is given in Listing 1-1.

Listing 1-1. QUICKSORT routine QUICK.DO.



```
9000 REM QUICKSORT PROGRAM FOR MODEL100


9001 REM SORT ARRAY IS X AND DIM(N) IN M
       AIN PROGRAM
9002 REM ST IS A STACK DIMENSION 2N/3
9003 REM VERSION 1.0 BY LES
9004 REM X AND ST DIMENSIONED IN MAIN
9005 REM ***************************
9010 JP=0
9020 JR=JP+JP
9030 ST(J+1)=1
9040 ST(J+2)=N
9050 JP=JP+1
9060 IF JP=0THENRETURN
9070 JP=JP-1
9090 J=JP+JP
9100 J1=ST(J+1)
9120 J2=ST(J+2)
9130 J3$=X$(J1)
9140 JZ=J1
9150 JB=J2+1
9160 JB=JB-1
9170 IF JB=JZ THEN 9230
9180 IFJ3$>X$(JB)THENX$(JZ)=X$(JB)ELSE91
       60
9190 JZ=JZ+1
9200 IFJB=JZ THEN 9230
9210 IFJ3$<X$(JZ)THENX$(JB)=X$(JZ)ELSE91
       90
```

```
9220 GOTO 9160
9230 X$(JZ)=J3$
9240 IFJ2-(JZ+1)<1THEN9280ELSEJR=JP+JP
9250 ST(JR+1)=JZ+1
9260 ST(JR+2)=J2
9270 JP=JP+1
9280 IFJB-1-J1<1THEN 9060 ELSE JR=JP+JP
9290 ST(JR+1)=J1
9300 ST(JR+2)=JB-1
9310 JP=JP+1
9320 GOTO9060
```

This particular sort is called QUICKSORT and is much faster than bubble sort techniques for most data. However, if the data are nearly in order, QUICKSORT can be slow. QUICKSORT requires an additional block of memory for a stack used in the sort. This stack should be dimensioned (2N/3) where N is the number of items to be sorted.

Type this program into your computer. Note that the line numbers were chosen to be outside the range normally used in programs. Now store the program in the RAM file with the command SAVE“QUICK.DO”,A. This command saves the program as an ASCII file that we can later merge with our program that uses the sort.

Now type in the rest of the program, as given in Listing 1-2. When you have it all typed in, type

Listing 1-2. Demonstration program for QUICKSORT.

```
1 DEFSNGA-Z
2 REM QUICKSORT DEMONSTRATION
3 REM VERSION 1.0 BY LES
4 REM FOR TAB BOOKS
10 CLS
20 DIM X(100),ST(2*100/3)
25 INPUT "number of items";N
40 FORJ=1TON:
    INPUTX(J):
   NEXTJ
50 FORJ=1TON:
   PRINTX(J),:
   NEXTJ
60 J=0:
   GOSUB9000
65 CLS
70 FOR J=1TON:
   PRINTX(J),:
   NEXTJ
80 STOP
```

MERGE"QUICK.DO". This will join our QUICKSORT routine to the program. The complete, merged program is shown in Listing 1-3 at the end of the chapter.

Now we have to do some housework on the sort routine to adapt it to the task at hand. The basic routine is designed to sort strings, but we want to sort numbers. So we have to remove all the dollar signs ($) from the variable names.

To do this type EDIT 9000—. As soon as the edit buffer is loaded, press function key F1 and, in response to the prompt STRING, type $ and press ENTER. The cursor will move to the first $, which we can delete. Press F1 again to remove the next $, and so on for the rest of the routine. Now notice that Lines 9001-9005 are general remarks and serve no purpose in our program. These can be removed by locating the cursor over the 9 in 9001 and then pressing F7,SELECT. Then move the cursor to the 9 of 9010 (a quick way to do this is press F1, type 9010 in response to the prompt, and then press ENTER). Now press F6, CUT, and the text is removed from the program. Now press F8 to get back to BASIC. Now run the program and debug it as necessary.

When you run the programs in the book, you'll discover that they reject incorrect responses. For example, if the computer asks for a Y or N answer to a question, it will accept no other response. This type of error checking is useful, and can be included in your programs. Error checking requires two functions from BASIC, INKEY$, and INSTR. INKEY$ is used to get a single key from the keyboard. The INSTR function is then used to determine if the key is valid. If it is valid, the program is allowed to continue. If the input is not valid, program control is returned to the point where the key should be pressed. An example should make this all clear.

Let's assume that we want to get a Y or N response to a question. The code for this follows:

```
10 A$= "YyNn"
20 PRINT"ARE YOU READY TO CON-
TINUE Y/N?";
30 Y$ =INKEY$ : IF Y$ = "" THEN 30
40 IF INSTR(A$,Y$ = 0 THEN PRINT
"ANSWER Y OR N":GOTO 30
50 REM NOW HAVE A Y OR N AND ARE
READY TO GO.
```

Line 10 of this example sets up the control string that contains the valid keys, Line 30 gets a character from the keyboard, and Line 40 checks to see if Y$ is contained in the valid string. If Y$ is not in A$, the value of INSTR(A$,Y$) is zero and we print a message to try again. Note that A$ was defined to allow both upper-and lowercase response.

Most of the programs use the INKEY$ function to get a single character response from the keyboard. The other way of getting such a response is with the INPUT$(1) command. Although these two commands do the same thing, they are different. The INKEY$ function checks to see if a key is pressed. If no key is pressed the program continues. If you want to wait for a response, you need code such as that shown in line 30 above. The INPUT$(1) command, on the other hand, stops program execution and waits for the response.

The main reason I used INKEY$ instead of INPUT$(1) is that INKEY$ does not print the cursor while it waits for a response. This means that the display is kept clean. INPUT$(1) prints the cursor to let you know that the computer is waiting for a response. The cursor can clutter up the display in ways I did not want in most of the programs.

Because of memory limitations in the Model 100, it is desirable to compress the programs as much as possible and to eliminate remarks. This means that a single line of BASIC may have many program statements separated by colons. While this does save memory, it also makes the programs harder to read and debug. To make it easier for you, the programs are listed as follows:

```
10 PRINT
PRINT"THIS IS AN EXAMPLE":
INPUT A$
```

This line should be typed as:

```
10 PRINT:PRINT"THIS IS AN EXAM-
PLE":INPUTA$
```

The programs are all designed to make the data input as much the same as possible. Thus all the programs use a full-display form of data entry. The screen is divided into rows and columns, with each useful row and each useful column named. You enter data by locating the cursor in the row and column corresponding to the datum you want to enter. You then type the datum and press ENTER. That's all there is to it. You can move the cursor up and down or right to left with the arrow keys.

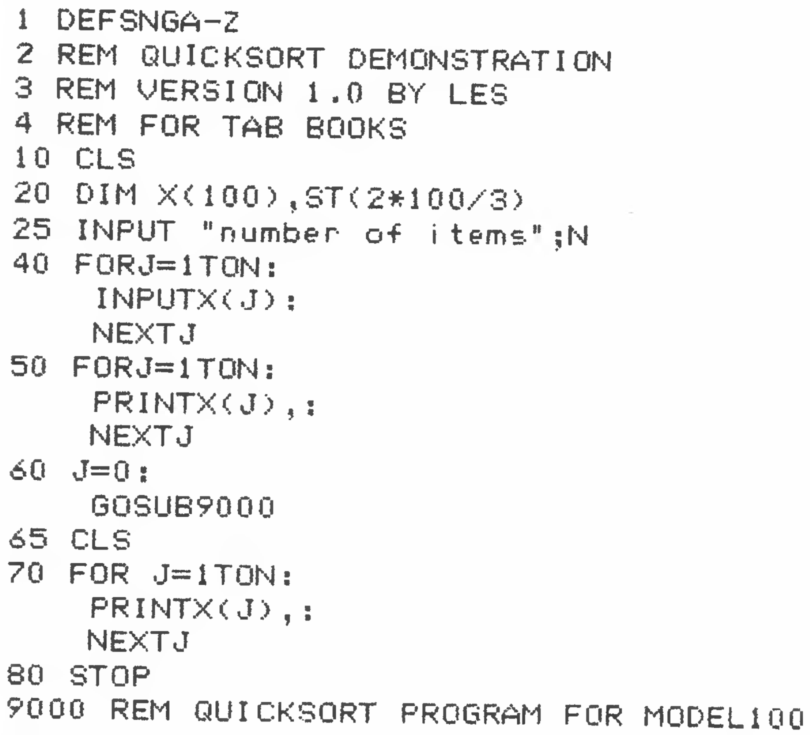
Note that in many cases there is no space on the small display to show all the input data. In this case the data are divided into "pages". You move from page to page with the shift arrow keys. The vertical keys change pages in the vertical direction and the horizontal arrow keys change pages in the horizontal direction. I think you'll find this a very easy way to enter your data.

When you have entered all your data, you start the calculations by pressing the asterisk (*)key. In some cases you can save data by pressing S. In other cases you can plot the data, and you do this by pressing P. Note that in all cases short instructions are displayed on the last line of the display. These tell you which keys are active and which keys to use for a given command. Also note that your input is checked to be sure that it is legal. For example, when you are entering numeric data, you can not enter letters. The program will not accept them.

In general the programs are self-contained and don't require much comment. The data entry form clearly defines what the computer wants in the way of information, so there is no need for much comment there. In fact, you should be able to run the programs with no instruction except for the instructions on the display.

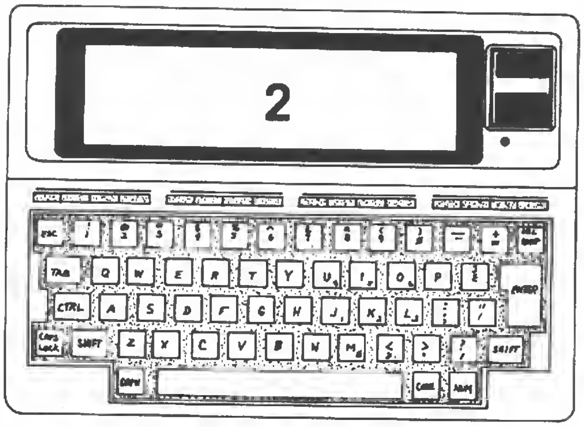
What I hope to accomplish in the text that goes with the programs is provide you with an understanding of what the purpose of the program is and what the concepts behind the program are. This way I hope that you will be in position to better use the programs. The computer *will* generate numbers no matter what; it is up to us, the users, to determine what the numbers mean.

Listing 1-3. Demonstration program and QUICKSORT merged.



```
1 DEFSNGA-Z
2 REM QUICKSORT DEMONSTRATION
3 REM VERSION 1.0 BY LES
4 REM FOR TAB BOOKS
10 CLS
20 DIM X(100),ST(2*100/3)
25 INPUT "number of items";N
40 FORJ=1TON:
     INPUTX(J):
     NEXTJ
50 FORJ=1TON:
     PRINTX(J),:
     NEXTJ
60 J=0:
     GOSUB9000
65 CLS
70 FOR J=1TON:
     PRINTX(J),:
     NEXTJ
80 STOP
9000 REM QUICKSORT PROGRAM FOR MODEL100
```

```
9001 REM SORT ARRAY IS X AND DIM(N) IN MA
        IN PROGRAM
9002 REM ST IS A STACK DIMENSION 2N/3
9003 REM VERSION 1.0 BY LES
9004 REM X AND ST DIMENSIONED IN MAIN
9005 REM ***************************
9010 JP=0
9020 JR=JP+JP
9030 ST(J+1)=1
9040 ST(J+2)=N
9050 JP=JP+1
9060 IF JP=0THENRETURN
9070 JP=JP-1
9090 J=JP+JP
9100 J1=ST(J+1)
9120 J2=ST(J+2)
9130 J3=X(J1)
9140 JZ=J1
9150 JB=J2+1
9160 JB=JB-1
9170 IF JB=JZ THEN 9230
9180 IFJ3>X(JB)THENX(JZ)=X(JB)ELSE9160
9190 JZ=JZ+1
9200 IFJB=JZ THEN 9230
9210 IFJ3<X(JZ)THENX(JB)=X(JZ)ELSE9190
9220 GOTO 9160
9230 X(JZ)=J3
9240 IFJ2-(JZ+1)<1THEN9280ELSEJR=JP+JP
9250 ST(JR+1)=JZ+1
9260 ST(JR+2)=J2
9270 JP=JP+1
9280 IFJB-1-J1<1THEN 9060 ELSE JR=JP+JP
9290 ST(JR+1)=J1
9300 ST(JR+2)=JB-1
9310 JP=JP+1
9320 GOTO9060
```



The Programs

Most of the programs are short and are designed to be kept in RAM storage. Most of the programs are designed to provide output to the display. A few do have provision for printing answers on a printer, but most of the programs are designed to only use the display. In situations where you want the calculated results printed, you can get a hard copy by pressing the PRINT key on the Model 100. This decision to limit output to the display was based on the assumption that most of the time you won't have a printer with you when you're using the Model 100. It is also based on the knowledge that the print function key can provide direct copy of the screen display if needed. Thus, except for one or two situations, there is no need for a separate line printer routine.

Most of the programs are designed to use a data entry format similar to that of a spreadsheet. The use of this format means that the data entry module of many of the programs is the largest module of the program. In most cases the data entry program is based on a master data entry module that is customized for each application program. The use of the master module approach significantly reduced the time spent developing and typing in the programs and ensured that the data entry procedures for all the programs were compatible.

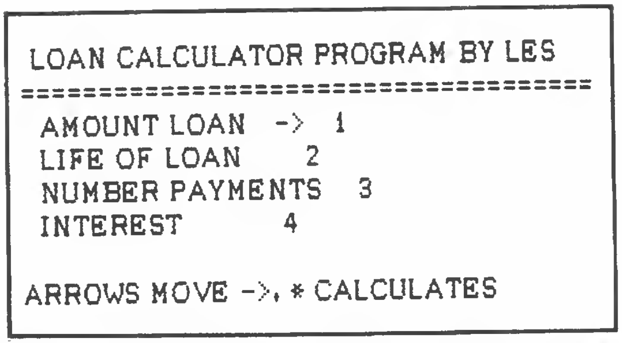
Because the master data entry module is so useful, I have provided it in Listing 2-1 at the end of this chapter. This module is designed to be merged with the calculation modules of an application program. Thus, once you type in the master data entry module, you do not have to retype the whole module for each program. All you have to do is make the modifications necessary to customize it for your application. I think you'll find this approach very effective.

Note that I have included several remarks in the listing of the master module. The remarks are there to help you figure out what is going on in the routine. Because remarks take valuable space in RAM, I suggest that you eliminate them when you type the program into your computer. The listings for the applications programs do not have all the remarks. You can type the programs in as listed and need not worry about removing remarks to save memory.

The data entry form uses two arrays to keep track of things. The string array T$()—or F$ in some of the programs—is used to keep track of the names of the variables; it is impossible to fill out a form if you don't know what information goes where. This array is filled in the subroutine beginning at line 9900. The subroutine is set up to read in the titles from a DATA statement. If the word END is encountered, the program knows that all the titles have been read in and it returns control to the main program. If END is not encountered, the string is stored in the appropriate location in the T$() array, the array index is incremented by one, and another title is read. The program could have been written to read the titles in using a FOR-NEXT LOOP. However, this would have required knowing the number of titles to be read beforehand. Since programs always have a way of evolving, the number of titles being read often changes. By making the computer keep track of how many titles are needed, I freed myself from a little work and helped make sure that the programs weren't likely to try reading too many or too few data.

The numerical array, X( ) is used to store the data that goes with each of the titles. One of the major sources of potential bugs when storing everything in arrays is that you can forget which variable is supposed to do what. For example, is X(1) the interest rate or the amount of the load? I know from experience that you can produce serious and very hard-to-find bugs if you forget what your variables are. The advantage of using arrays instead of meaningful data names, at least in the case of the Model 100, outweigh the disadvantages—especially if you take care and make a variable list.

The data entry module can even provide the format for making the variable list. Instead of entering values for the variables, just enter the index for the variable. For example, assume that you want to work with the following items to calculate a loan schedule:



```
 LOAN CALCULATOR PROGRAM BY LES
========================================
  AMOUNT LOAN  ->  1
  LIFE OF LOAN     2
  NUMBER PAYMENTS   3
  INTEREST       4

 ARROWS MOVE ->, * CALCULATES
```

Fig. 2-1. Using the data entry module to compile a variable list.

AMOUNT OF LOAN= X(1)
LIFE OF LOAN= X(2)
NUMBER OF PAYMENTS= X(3)
INTEREST RATE= X(4)

The data statement in line 9900 is then set to

```
9900 DATA AMOUNT OF LOAN, LIFE OF
LOAN,NUMBER OF PAYMENTS, INTER-
  EST,END
9910 J = 1
9920 READ T$:IF T$ = "END"THEN RE-
  TURN ELSE T$(J) = T$
9930 J = J + 1
9940 GOTO 9920
```

You then use the data entry module. In the row for Amount of Loan, enter 1. In the row for Interest Rate, enter 2, and so on. When you have entered all the numbers 1, 2, 3, and 4, press PRINT to get a hard copy of the display. You then get the printout in Fig. 2-1. This printout is a complete listing of your array variables. Now when you write the rest of the program to use the data, just refer to the printout to find out what you use when you need the amount of the loan. You'll find that now you don't have to depend on your memory.

PROGRAM STRUCTURE AND DESIGN

The discussion of how to use the data entry module in your programs will be clearer if we spend a little time discussing program structure in general. The general ideas in this section can be applied to your own programs and will make you a more productive programmer.

The first thing to remember is that good programming does not depend on the language in which you program. You can write structured programs in BASIC and unstructured programs in Pascal. It is how you use the language that matters, `not the language itself.

You should organize your programs in blocks or *modules*. Each module has a specific function and may have submodules associated with it. Each module can be debugged by itself and later merged into the main program. The advantage of using modules is that, if a change needs to be made in a module, you can confine the changes to that module and then merge the modified module back into the program. Since you only changed one module, you know that the rest of the program is still correct. Any errors that crop up in the modified program must be due to errors in the module you changed. Some of the modules you might have in a program are shown in Fig. 2-2. Each module is discussed briefly below.

The first module is the *identification* module. The purpose of this module is to identify the program. It tells the program name, the version, the author, and perhaps some additional information. The module is mostly remark statements, although it might contain a print message to display some of the important identification information. This module is often forgotten—especially if the program is for your use and not for others. But it should not be forgotten. it is not uncommon to have copies of several versions of the same program lying around; The identification module is the only way to find out which version is the current one and the one you want to use.

The next module is the *initialization* module. The purpose of this module is to set up everything for use by the rest of the program. If variables have to be set to some value, this module does the work. This is the module that takes care of filling the title arrays.

The next module is the *data entry* module. The bulk of the data required to use the program should be entered here. This is the module that the user will see the most. Nearly all the user's interaction with your program takes place here. If this module isn't user-friendly, the user won't care how user-friendly the rest of the program is. (Spreadsheet data entry or fill-in-the-forms data entry is about as user-friendly as you can get.)

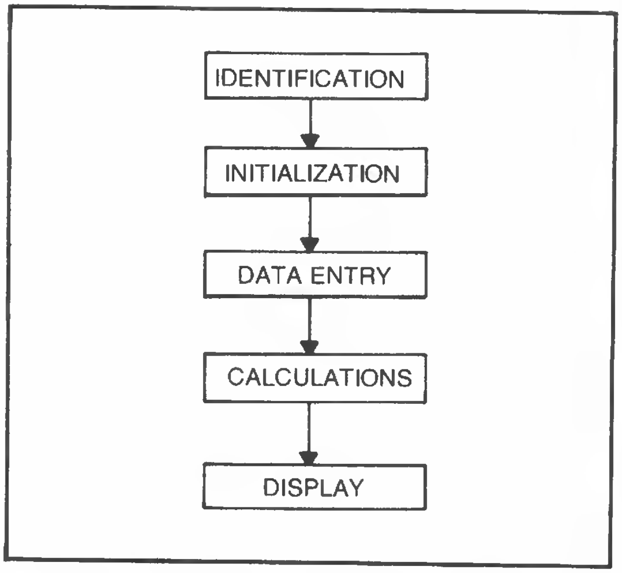


Fig. 2-2. Block diagram of modular program structure. All modules are called from the main program.

The next module is the *calculation* module. Here is where the computer does its thing. Then there is the *display* module. This can be divided into two submodules—one for display on a screen and one for hard copy. The modules are called from the main program as subroutines, and the modules can also have their own subroutines.

USING THE DATA ENTRY MODULE

To use the master module, first type it into your computer using BASIC and then debug it. Now that it is working save it in RAM as an ASCII file with the command SAVE"DATIN.DO",A. Note that you are saving the program as a *document* (or .DO) file and not as a BASIC or .BA file. This means that the program will take more RAM and that the program can be merged.

The data entry module requires that code be added to the initialization module to set up the title array and to set up the screen format. The statements to initialize the data entry module are in the subroutine beginning at line 9900. Thus to call the initialization routines from the main program, you must add the statement GOSUB 9900. As discussed previously, the subroutine starting at 9900 fills the title array, and also sets up the strings that determine what kind of input is allowed.

Note the two string variables A$ and AZ$. These two strings are used to insure that only acceptable input is entered. In all the programs in the book AZ$ limits input to the digits 0 to 9, the plus sign, the minus sign, the period, the back arrow (allowed for corrections), and the ENTER keys. When the program is looking for numeric input, it uses AZ$ to filter out all other keys. If an incorrect key is pressed, nothing happens. The A$ string is used to limit input to numeric information and the cursor and page changing keys.

Back to your program. You have everything initialized and are now ready to enter data. So type GOSUB 9000 to transfer control. The main program is now ready for DATIN module. Type MERGE"DATIN.DO" and press ENTER. The word WAIT will be displayed and the module loaded. Note that any statements you have with the same line numbers as are used in the data entry module will be replaced with lines from DATIN.DO. So do not use line numbers 9000-9999 in your main program.

As soon as the program has been merged, type in the appropriate data statements as part of the subroutine at 9900. Now you're ready to go. Try using the module for a few of your programs. I think that you'll soon find that it is better and easier to use than the method you're now using.

GENERAL COMMENTS

Although the Model 100 is designed to use RAM as storage, you should have a backup copy of the data entry module and the rest of your programs on tape somewhere—just in case. The first step is to hook up your cassette recorder as shown in the manual. Put a cassette tape in the recorder, fast forward the leader out of the way, and then check to see that actual tape is located over the recording head and not leader. Load DATIN.DO into BASIC and wait until the computer is ready. Save the program with the CSAVE"DATIN command.

When the recorder has stopped, rewind the tape to the beginning and use the CLOAD? command to verify that you got a good save. If you didn't, try to verify the recording with the volume level increased. If this doesn't work, resave the program and try again. If you still can't get a good verify, clean the heads of the recorder and try one more time.

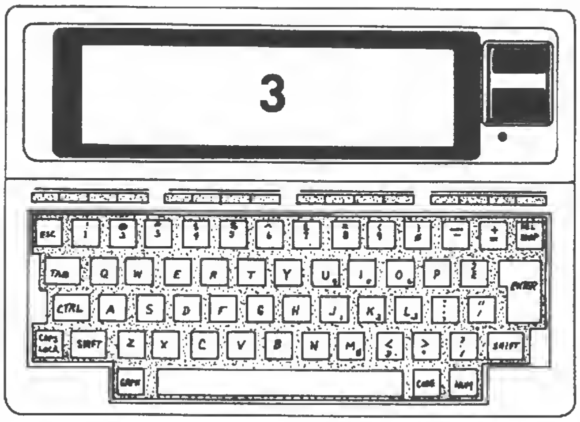
As soon as you have a good save, place the tape in a safe place—and hope that you never need it.

Listing 2-1. Master data entry module.

```
9000 REM DATA INPUT
9001 REM MASTER DATA ENTRY MODULE
9002 REM VERSION 1.0 BY LES SEP83
9003 REM VERSION OF ENTERING X,Y DATA
9004 REM INITIALIZED IN SUBROUTINE
9005 REM STARTING AT 9900
9006 REM CALL FROM MAIN BY GOSUB9900
9007 REM TO INITIALIZE AND THEN BY
9008 REM GOSUB 9000
9009 REM DO NOT USE LINENUMBERS 9000-9999
     IN MAIN PROGRAM.
9010 CLS:
   PRINT "NO.";TAB(12)X$;TAB(26);Y$
9020 PRINT STRING$(39,"=")
9040 FOR J=J1TOJ2
9050    PRINT J;TAB(8);X(J);TAB(25);Y(J)
```

```
9060 NEXT J
9070 PRINT @(JS*40)+JR,"->";
9075 PRINT@280,"ARROWMOVE>SHIFTARROWCHANGE
     -99LASTX*RET";
9080 Y1$=INKEY$:
   IFY1$=""THEN9080
9090 IFINSTR(A$,Y1$)<>0 THEN 9500
9100 JZ=INSTR(AZ$,Y1$):
   IFJZ=0THEN9080
9110 ON JZ GOTO 9120,9120,9140,9160,9180,9
     199,9200,9250
9120 PRINT@(JS*40)+JR,"   ";
9122 JS=JS+1:
   IFJS>6THEN JS=2:

9124 PRINT@(JS*40)+JR,"->";
9130 GOTO 9080
9140 PRINT@(40*JS)+JR,"   ";
9150 JS =JS-1:
   IFJS<2 THEN JS=6
9152 PRINT@(40*JS)+JR,"->";:
   GOTO9080
9160 PRINT@(JS*40)+JR,"    ";
9170 IF JR=4 THEN JR=22 ELSEJR=4
9175 PRINT@(JS*40)+JR,"->";:
   GOTO 9080
9180 PRINT@(JS*40)+JR,"  ";
9185 IF JR=22 THEN JR=4 ELSE JR=22
9190 PRINT@(JS*40)+JR,"->";:
   GOTO 9080
9199 RETURN
9200 J1 = J2+1:
   J2=J1+4
9210 JR = 4:
    JS=2
9220 GOTO 9010
9250 IF J1 = 1 THEN 9010
9260 J2 = J1-1:
   J1 = J2 - 4
9270 GOTO 9010
9500 PRINT @(40*JS)+JR,"   ";
9505 PRINT@280,"ENTER REST OF DATA PRESSEN
     TER WHEN DONE";
9510 V$=Y1$
9530 PRINT@(40*JS)+JR,V$;CHR$(95);"   ";
```

```
9540 Y1$=INKEY$:
   IF Y1$ =""THEN 9540
9550 IF Y1$=CHR$(13) THEN 9700
9560 IF Y1$<>CHR$(29) THEN 9600
9570 IF LEN (V$)=0 THEN 9530
9580 V$=LEFT$(V$,LEN(V$)-1):
9590 Y1$="":
   GOTO 9530
9600 IF INSTR(A$,Y1$)=0 THEN 9530
9610 V$ = V$ + Y1$
9620 GOTO 9530
9700 IF JR = 4THEN X(J1+JS-2)=VAL(V$):
   IFX(J1+JS-2)=-99THENN=J1+JS-3:
9701 IFJR=4THENJR=22:
   GOTO9010
9710 Y(J1+JS-2)=VAL(V$):
   JS=JS+1:
   JR=4
9720 IF JS>6 THEN JS=2
9730 GOTO 9010
9900 REM INITIALIZATION ROUTINES
9951 A$="1234567890.+-"
9952 AZ$=CHR$(13)+CHR$(31)+CHR$(30)+CHR$(2
     8)+CHR$(29)+CHR$(42)+CHR$(2)+CHR$(2
     0)
9953 J1=1:
   J2 = 5
9954 JS=2:
   JR=4
9960 REM DATA STATEMENTS FOR STRINGS GO HE
     RE
9970 REM THE LAST ITEM IN THE DATA LIST
9980 REM MUST BE END
9982 DATA END
9985 J=1
9990 READ T$:
   IFT$="END"THEN RETURN ELSE T$(J)=T$
9992 J=J+1:
   GOTO9990
```



Time and Money

The relationship between time and money is fundamental to financial decision making. You must understand this relationship in order to understand the economic consequences of your financial decisions; correct financial decisions are not possible without it. Once you do understand the relation between time and money, you can analyze all kinds of financial situations, including insurance, retirement, loans, acquisitions of equipment and real estate, and almost every other financial situation you are likely to encounter.

When you have a dollar, you can either spend it or you can invest it with the idea of spending it later. If you invest it, you are giving up the immediate purchase of goods and services. In order to justify this sacrifice, you expect to earn a profit on the investment. That is, you expect to get your dollar back plus a little extra to make up for the loss of the goods and services you could have purchased now. This profit is commonly called *interest,* and it is interest that makes people give up the immediate use of their money in hope of future gain.

The *time value of money* is important when you want to borrow or loan money. We will look at both situations and provide programs for the Model 100 that will carry out all the calculations necessary to make rational financial decisions.

COMPOUND INTEREST

There are two kinds of interest, *simple* and *compound.* With simple interest only the initial amount invested earns the interest. If we invest $100 at 10 percent per year simple interest, we earn $10 annually. At the end of one year we have $110, at the end of two years we have $120, and so on.

With compound interest, *both* the interest earned *and* the initial investment earn interest. For example, if we invest $100 at 10 percent compounded annually, we earn $10 in interest the first year (10 percent of $100). In the second year we earn interest on $110, or $11, in the third year we earn interest on $121 (or $12.10), and so on to the end of the investment period.

As you can see, your money grows much faster with compound interest than with simple interest. The math necessary to determine the results of compound interest is more difficult than that for simple interest—but not *that* much more difficult. With a computer the difference in difficulty of calculating the results doesn't matter.

The formula for calculating the results of compound interest can be derived fairly easily. Because the entire concept of time value of money depends on understanding compound interest calculations, we will work out the equation one step at a time.

Suppose you want to invest $1,000 at 10 percent interest per year compounded annually. You plan to leave the money for five years and want to know how much money you will have at the end of that time. We will work the problem one year at a time, but first let's define some terms:

PV (present value) = the value of your investment now ($1,000)
FV (future value) = the value of your investment at some future time
r = the interest rate expressed as a fraction (0.01 in our example)
n = the number of compounding periods (in this case, 5 years)

The future value of our $1,000 at the end of the first year is given by

$$\begin{aligned} FV1 &= PV + PV \times r \\ &= PV\ (1 + r) \\ &= 1{,}000 \times 1.1 \\ &= 1{,}100 \end{aligned}$$

For the second year we calculate using FV1

$$FV2 = FV1 + FV1 \times r = FV1\ (1 + r)$$

But FV1 = PV(1 + r), so we can substitute that expression for FV1

$$FV2 = PV(1 + r)(1 + r)$$

The third year we again substitute using PV(1 + r)

$$FV3 = FV2(1 + r) = PV(1 + r(1 + r)(1 + r)$$

We can now generalize the formula as

$$FVn = PV(1 + r)^n$$

We can use this formula to answer our question.

$$FV5 = 1{,}000(1.1)^5 = \$1{,}610.51$$

If we had invested at simple interest, we would have earned only $1,500.

It should be clear that we can solve the equation for any one variable if we know all the others. For example, the present value of a future amount is given by:

$$PV = FV(1 + r)^{-n}$$

Which can also be expressed as $PV = \dfrac{FV}{(1+r)^n}$. The present value formula is important because it tells us how much a given future value is worth today. In other words, the present value tells us how much we should be willing to pay now for the right to receive a given future amount.

We have been assuming that the compounding period was one year. The compounding period, however, can be any time period—daily, weekly, monthly, quarterly, and so on. The interest rate is generally stated as a percentage per year. To use the formula above we both divide the yearly interest rate and multiply the number of years by the number of compounding periods per year. The formula thus becomes

$$PV = FV(1 + r/q)^{-nq}$$

where q is the number of compounding periods per year.

For example, if we are offered 10 percent interest per year compounded monthly, we divide 0.1 by 12 to get the interest rate per compounding period, and multiply 3 years by 12 to get the total number of compounding periods. (Note that the programs in this book require only that you enter the yearly interest rate and the number of compounding periods per year. The computer does all the calculations for you.)

Since the computer can be made to do all the calculations, let's look at the computer program and run several examples, which will clear up any confusion that might exist.

PROGRAM COMPND.BA

The Model 100 program COMPND.BA (Listing 3-1) does all the calculations to solve compound interest problems. It is easy to use and quickly calculates the answer. To use the program you simply enter the variables you know and let the program calculate the missing one. If you enter too many variables, the program will tell you so, and if you enter too few it will also tell you. You tell the program that you want it to start calculating by pressing the * key.

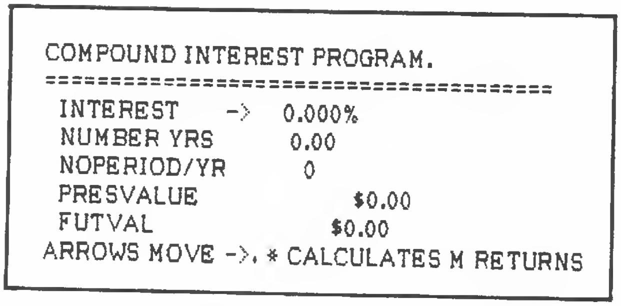
The following variables are present in compound interest problems:

- The present value (the value today)
- The yearly interest rate
- The number of compounding periods per year
- The number of years
- The future value (the value at a given future time)

The program can calculate any of the variables, provided all the others are known.

The beginning screen for the program is shown in Fig. 3-1. Note that the cursor, shown as the > symbol, points to the line where data entry will take place. To enter data simply enter the numbers. When you have entered all the numbers, press ENTER. The screen will display all the data you entered and the cursor will move to the next line. You can enter data for any line by moving the cursor up or down using the arrow keys.

Note that the cursor "wraps" around the screen. That is, if you press the down arrow when the cursor is at the bottom of the display, the cursor moves to the top of the display. When you press the up arrow with the cursor at the top, it moves to the bottom of the display.



```
COMPOUND INTEREST PROGRAM.
=====================================
 INTEREST    ->  0.000%
 NUMBER YRS      0.00
 NOPERIOD/YR      0
 PRESVALUE            $0.00
 FUTVAL            $0.00
ARROWS MOVE ->, * CALCULATES M RETURNS
```

Fig. 3-1. Beginning screen for compound interest program.

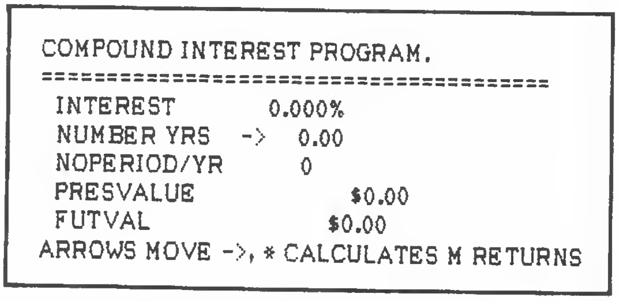
You enter zero (0) for the variable that you wish to calculate. The program requires that one, and only one, of the variables be equal to zero. When you have entered all the data, press * to start the calculations. If none or all of the variables are equal to zero, the program prints an error message and returns you to the data entry form.

Let's run a few examples to show what the program can do.

Example 1

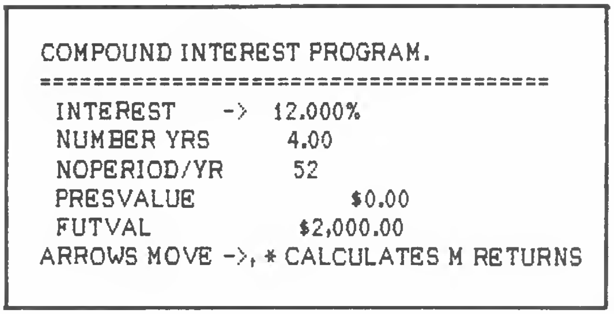
Suppose you know that you will need $2,000 four years from now. Your bank will give you a certificate of deposit paying 10 percent interest per year, compounded weekly. You want to know how much money you need to deposit now, in order to have your $2,000 at the end of four years. We know that future value = 2000, number of years = 4, interest rate per year at 10 percent, and the number of compounding periods per year is 52. Present value is the unknown.

We run COMPND.BA and get the display shown in Fig. 3-2. Because the present value is the



```
COMPOUND INTEREST PROGRAM.
=====================================
 INTEREST        0.000%
 NUMBER YRS  ->  0.00
 NOPERIOD/YR      0
 PRESVALUE            $0.00
 FUTVAL            $0.00
ARROWS MOVE ->, * CALCULATES M RETURNS
```

Fig. 3-2. Initial display for calculating Example 1.



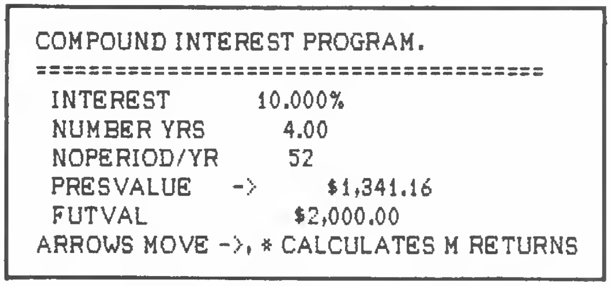
```
COMPOUND INTEREST PROGRAM.
========================================
 INTEREST    ->  12.000%
 NUMBER YRS       4.00
 NOPERIOD/YR      52
 PRESVALUE               $0.00
 FUTVAL              $2,000.00
ARROWS MOVE ->, * CALCULATES M RETURNS
```

Fig. 3-3. Completed data entry for Example 1.

unknown, we enter 0. The cursor now moves to interest rate, and we enter 10. We progress through the data entry screen until all the data are entered. The display should look like Fig. 3-3 when all the data are entered. Review all the data to be sure they are correct. Then press * to start the calculations. The display for the answers is shown in Fig. 3-4. The program is now ready for data entry, so another problem can be worked.

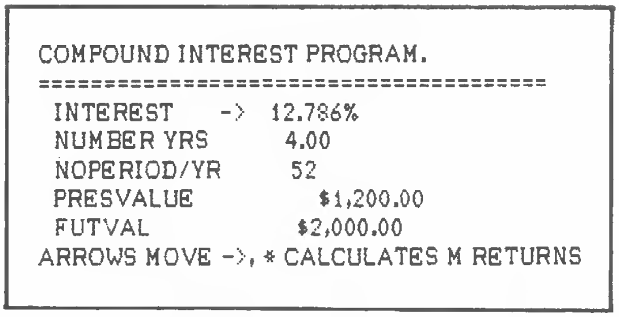
Let's assume that we don't have $1,341, but that we do have $1,200 to invest. What interest rate must we have to get the $2,000? In this case the interest rate is unknown, so we enter the known present value ($1,200), enter 0 to show that the interest rate is unknown, and leave everything else the same. Review the data to be sure they are correct and press the * key. The display in Fig. 3-5 shows the answer.

Note that the program does not provide for hard copy output. If you want hard copy, simply press the PRINT key on the Model 100. This will provide you with an exact copy of the display.



```
COMPOUND INTEREST PROGRAM.
========================================
 INTEREST        10.000%
 NUMBER YRS       4.00
 NOPERIOD/YR      52
 PRESVALUE   ->       $1,341.16
 FUTVAL              $2,000.00
ARROWS MOVE ->, * CALCULATES M RETURNS
```

Fig. 3-4. Calculated results for first part of Example 1.



```
COMPOUND INTEREST PROGRAM.
========================================
 INTEREST    ->  12.786%
 NUMBER YRS       4.00
 NOPERIOD/YR      52
 PRESVALUE            $1,200.00
 FUTVAL              $2,000.00
ARROWS MOVE ->, * CALCULATES M RETURNS
```

Fig. 3-5. Calculated results for second part of Example 1.

Example 2

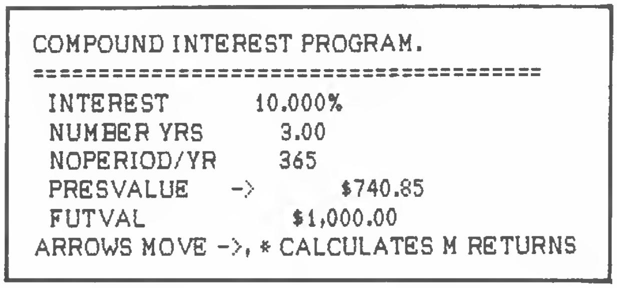
You wish to have $1,000 at some point in the near future. After careful study you narrow your choices to two. Investment A pays 10 percent compounded daily, while Investment B pays 10.1 percent interest compounded monthly. Both investments have a life of three years. Which is the better investment? Run the program for Investment A first and get the results shown in Fig. 3-6. Note that the present value is $740.85.

Now rerun the example for Investment B. Change the interest rate to 10.1, the number of compounded periods to 12, and the present value to 0. Press * and you get the display shown in Fig. 3-7.

Investment B is the better of these two investments because the present value is lower.

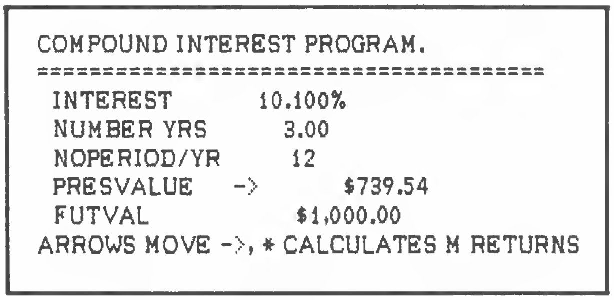
There are some combinations of input data that do not give a sensible answer. For example, assume that you try to find the number of compounding periods per year necessary to turn $2,000 into $5,000 at the end of two years, with an interest rate of 4 percent per year. There simply is no way for this to happen. The program will try to find the answer and will print an error message when it discovers that no solution exists.

Another example, suppose you need $2,000 in two years and have $2,100 to invest. Since the



```
COMPOUND INTEREST PROGRAM.
========================================
 INTEREST        10.000%
 NUMBER YRS       3.00
 NOPERIOD/YR      365
 PRESVALUE   ->        $740.85
 FUTVAL              $1,000.00
ARROWS MOVE ->, * CALCULATES M RETURNS
```

Fig. 3-6. Results for Investment A of Example 2.



```
COMPOUND INTEREST PROGRAM.
=======================================
 INTEREST       10.100%
 NUMBER YRS       3.00
 NOPERIOD/YR      12
 PRESVALUE   ->         $739.54
 FUTVAL              $1,000.00
ARROWS MOVE ->, * CALCULATES M RETURNS
```

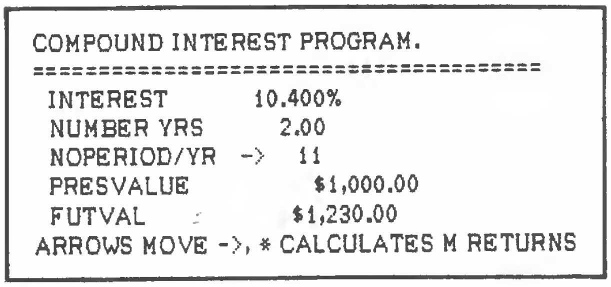
Fig. 3-7. Results for Investment B of Example 2.

present value is greater than the future value, the required interest rate is negative, i.e., you must lose money. Again the program will print a message telling you that there is no sensible answer.

Some of the variables are calculated using a trial-and-error procedure. When the calculated answer is within a set tolerance, the calculation stops. Thus it is possible that you will find some small errors in the calculations. This is especially true for the case of solving for the number of compounding periods per year. The program calculates the nearest integer number of periods (the number of periods is a whole number) that will give the desired future value. If you first calculate the number of compounding periods, and then if you calculate the future value for the calculated number of compounding periods as a check, you will often find that the calculated future value is not the future value you started with. This is a consequence of limiting the number of compounding periods to whole numbers. Perhaps an example will help.

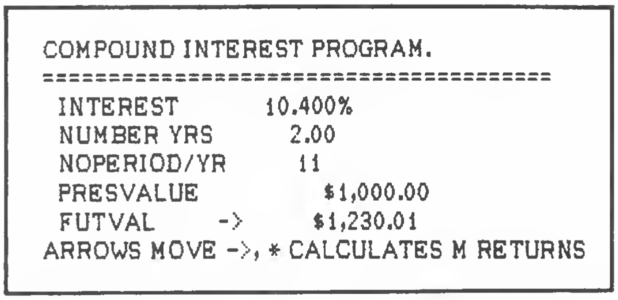
Example 3

Assume that you have $1,000 and want a future value of $1,230 in two years. You can earn 10.4 percent interest per year and you want to know how many compounding periods per year it will take to give you the money. Run COMPND.BA and set number of periods to 0. The calculated answer is shown in Fig. 3-8. Now, as a check on the program, set the future value to 0 and recalculate. The answer is shown in Fig. 3-9. Note that the calculated future value is not the same as the future value you started with. This is not a error in the program, but rather a consequence of limiting the number of periods to integer numbers.



```
COMPOUND INTEREST PROGRAM.
=======================================
 INTEREST       10.400%
 NUMBER YRS       2.00
 NOPERIOD/YR  ->  11
 PRESVALUE            $1,000.00
 FUTVAL              $1,230.00
ARROWS MOVE ->, * CALCULATES M RETURNS
```

Fig. 3-8. Calculated number of periods for Example 3.



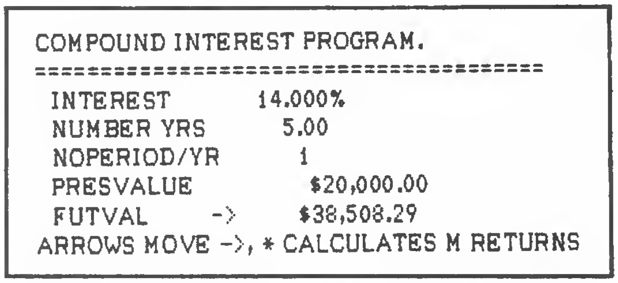
```
COMPOUND INTEREST PROGRAM.
=======================================
 INTEREST       10.400%
 NUMBER YRS       2.00
 NOPERIOD/YR      11
 PRESVALUE            $1,000.00
 FUTVAL      ->      $1,230.01
ARROWS MOVE ->, * CALCULATES M RETURNS
```

Fig. 3-9. Crosscalculation of future value for Example 3. Note the discrepancy with Fig. 3-8.

Let's run a few more examples to show the kinds of questions that COMPND.BA can answer.

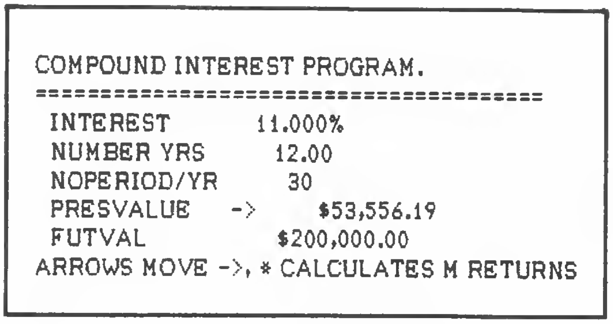
Example 4

Assume you are offered the chance to purchase property for $20,000. Your studies show that the property should appreciate at 14 percent per year, compounded annually. You want to sell the property in five years and want to know what the property will sell for. Since this is a future value problem, run the program and set future value to 0. The answer is shown in Fig. 3-10. Note that the property will sell for $38,508.



```
COMPOUND INTEREST PROGRAM.
=======================================
 INTEREST       14.000%
 NUMBER YRS       5.00
 NOPERIOD/YR       1
 PRESVALUE            $20,000.00
 FUTVAL      ->      $38,508.29
ARROWS MOVE ->, * CALCULATES M RETURNS
```

Fig. 3-10. Calculated future value for Example 4.



```
COMPOUND INTEREST PROGRAM.
=====================================
 INTEREST        11.000%
 NUMBER YRS       12.00
 NOPERIOD/YR        30
 PRESVALUE    ->       $53,556.19
 FUTVAL           $200,000.00
ARROWS MOVE ->, * CALCULATES M RETURNS
```

Fig. 3-11. Calculated present value for Example 5.

Example 5

You plan to retire in 30 years and want to have $200,000 available in savings. Your employer offers to make a lump sum deposit that will give you that amount. The interest rate is 11 percent per year compounded monthly. How much must your employer deposit?

This is a present value problem. You know the interest rate, the number of compounding periods per year, the number of years, and the future value, so enter all the knowns and set present value to 0. The results of the calculations are shown in Fig. 3-11.

Example 6

You know that in 10 years you will need $90,000. You have $40,000 available to invest to give you this amount. What is the interest rate required to give you $90,000 with annual compounding? The answer is shown in Fig. 3-12.

As you can see from the example, COMPND.BA gives you the capability of comparing lump sum payments that occur at different points in time. Without the program and the concepts discussed above, it is hard to know if it is better to receive $5,000 now or $10,000 in eight years. The comparison can be made by comparing the values of the two payments at a common point in time. You can either determine the future value of the $5,000 or the present value of $10,000. Once you have decided which point in time you want to use for the comparison, you need to determine the interest rate that is available to you. Once you know the interest rate, you can run COMPND.BA and find which investment you should select.

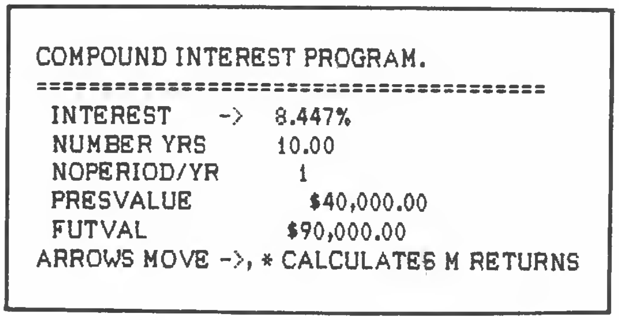
As a drill, you might want to run COMPND.BA and see which is better: $5,000 now or $10,000 in eight years if the interest rate is 10 percent annually. Now see what happens if the interest rate is 10 percent compounded monthly.

COMPOUND INTEREST WITH PERIODIC PAYMENTS

Although there are many cases where we are interested in lump sum payments, it is more common to be interested in the effects of time on periodic payments. For example, we may want to know how much we will have in five years if we can save $100 per month. Or we may want to know how much we can withdraw from savings each month if we want the savings to last four years. These are examples of *annuity* problems.

An annuity is defined as a payment (or receipt), usually of a fixed amount, made at stated, predetermined intervals. Most commonly the payment is made at the end of the payment period and is called an *ordinary annuity*. If the payment is made at the beginning of the period, the annuity is called an *annuity due*. We will look at both types of annuities. Two programs, ANUIT1.BA and ANUIT2.BA are included to provide solutions to both types of problems.

In order to fix the concept, let's work through an example and then derive the formulas. Suppose we can set $1,000 aside at the end of each year for the next five years. The money will earn 6 percent interest per year compounded annually. How much will we have at the end of the fifth year? At the end of the first year $1,000 is deposited and earns 6 percent interest for the next year. Also at the end of the second year we deposit the second $1,000. The



```
COMPOUND INTEREST PROGRAM.
======================================
 INTEREST      ->  8.447%
 NUMBER YRS        10.00
 NOPERIOD/YR         1
 PRESVALUE             $40,000.00
 FUTVAL              $90,000.00
ARROWS MOVE ->, * CALCULATES M RETURNS
```

Fig. 3-12. Calculated interest rate for Example 6.

Table 3-1. Example of Ordinary Annuity.

ORDINARY ANNUITY

$1,000 per year for 5 years at 6% interest

End of Year	*Deposit	Interest Earned	Total Value
1	$1,000.00	$ 0.00	$1,000.00
2	1,000.00	60.00	2,060.00
3	1,000.00	123.60	3,813.60
4	1,000.00	191.02	4,374.62
5	1,000.00	262.48	5,637.10
Totals	$5,000.00	$ 637.10	$5,637.10

* Deposit made at end of each year

value of our account at the end of the second year is $1,000 (initial deposit) + $60 (interest on the first $1,000) + $1,000 (the second deposit) = $2060. At the end of the second year we deposit the third $1,000 and so on. The results are summarized in Table 3-1.

We can use the same example to demonstrate the difference between an ordinary annuity and an annuity due. Remember that ordinary annuity payments are made at the end of the period; payments for an annuity due are made at the beginning of the period. Thus, if our example problem were for an annuity due, we would make the first deposit at the beginning of the first year. At the end of the first year we would have $1000+ $60 (the interest earned). At the end of the second year we would have $1000 (the first deposit)+ $60 (the interest earned on the first $1000)+ $1000 (the second deposit) + 123.60 (The interest earned during the second year on the $2060). The result for an annuity due are shown in Table 3-2.

These two examples show that the difference between an ordinary annuity and an annuity due is one year's worth of interest. An annuity due earns one more year's interest than an ordinary annuity.

Although it is possible to work any annuity problem this way, it is easier to use the annuity formulas. For an annuity due

$$FV1 = S[r((1 + r)^n - 1)][1 + r]$$

where FV1 is the future value of the annuity due (a lump sum), S is the amount of the payment, r is the interest rate per period, and n is the number of periods. The equation for the present value of an annuity due is

$$PV1 = S([r(1 + r)^n)][1 + r]$$

where PV1 is the present value of the annuity due. This equation says that if we set aside PV1 dollars now at an interest rate of r per period, we can withdraw S dollars per period for n periods.

The equations for an ordinary annuity are:

$$FV= S \frac{(1 + r)^n - 1}{r}$$

where FV is the future value of the ordinary annuity and all the other terms are as defined for the annuity due. The present value of an ordinary annuity is given by:

$$PV= S \frac{1 - (1 + r)^{-1}}{r}$$

where PV is the present value of an ordinary annuity.

There are two types of annuity problems. The first is the future value problem, which (obviously) uses the future value formulas for solution. The future value problem involves a lump sum that you

Table 3-2. Example of Annuity Due.

ANNUITY DUE

$1,000 per year for 5 years at 6% interest

End of Year	*Deposit	Interest Earned	Total Value
1	$1,000.00	$ 60.00	$1,060.00
2	1,000.00	123.60	2,813.60
3	1,000.00	191.02	3,374.62
4	1,000.00	262.48	4.637.10
5	1,000.00	338.23	5,975.33
Totals	$5,000.00	$975.33	$5,975.33

*Deposit made at beginning of each year

will receive in the future, provided that you set aside a given sum at regular intervals. Generally, you either know the lump sum you need and want to know what the regular savings have to be, or you know how much you are setting aside and want to know the value of the lump sum.

The other general problem is the present value problem. In this case you have a lump sum now and want to receive it in the form of regular payments for some time in the future. To solve this type of problem, you use the present value formulas.

Quite often we are interested in both type of problems at the same time. A retirement fund is an example of a case where we may have both a present value and a future value problem: as long as we are paying into the retirement fund, we have a future value problem, but if we want to know how much we can withdraw from the retirement fund, we have a present value problem. The two programs ANUIT1.BA and ANUIT2.BA can be used to solve both present value and future value problems. ANUIT1.BA is for annuity due problems (where the payments are made at the beginning of the period) and ANUIT2.BA is for ordinary annuity problems. The two programs operate in identical fashion and require the same information to run.

Because ordinary annuities are the more common form of annuity, the remaining discussion will cover ANUIT2.BA.

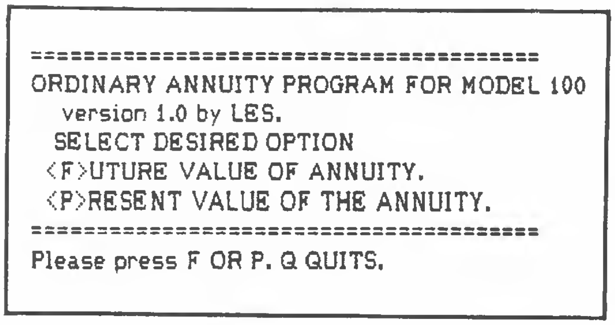
As soon as you run ANUIT2.BA you see the display shown in Fig. 3-13. The program needs to know if you are working a future value or a present value problem. Answer the question by pressing P for present value or F for future value. If you want to quit and return to BASIC, press Q. If you press any other key, the program will reject your answer, ask you to press either F, P, or Q and then wait for your response.

As soon as you answer the question (in this case for a future value problem) the computer displays the data entry form Fig. 3-14. This is the standard data entry form used by most of the programs in this book. You move from item to item by pressing the up and down arrow keys. You can also move to the next item by pressing the RETURN key.

Enter the data for the items that you know and enter 0 for the one item for which you wish to solve. As soon as all the data are entered, press the * key to start calculations. If you entered 0 for more than one item, an error message will be displayed and you will be returned to the data entry form to fill in the missing information. If you entered data for all the items, you will also get an error message and will be returned to the data entry form to enter 0 in the appropriate place. While the program is calculating the answer, it displays the item that it is calculating.

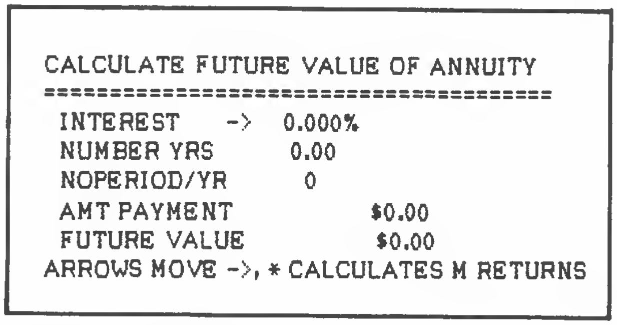
When the calculations are complete, the answer is displayed in the data entry form. The > is located so that it points to the item calculated. If you want to print the results, simply press the PRINT key on the Model 100. This will print the display on the printer. The advantage of doing the print this way is that all the input data and the calculated answer are shown. Some time later you will be able to figure out what problem you were working on and not have to depend on remembering the data.

If you wish to do additional calculations, simply enter the new data. Be sure that you enter 0 for



```
======================================
ORDINARY ANNUITY PROGRAM FOR MODEL 100
  version 1.0 by LES.
 SELECT DESIRED OPTION
 <F>UTURE VALUE OF ANNUITY.
 <P>RESENT VALUE OF THE ANNUITY.
======================================
Please press F OR P. Q QUITS.
```

Fig. 3-13. Introductory display for ordinary annuity program.



```
CALCULATE FUTURE VALUE OF ANNUITY
======================================
 INTEREST    ->  0.000%
 NUMBER YRS      0.00
 NOPERIOD/YR      0
 AMT PAYMENT             $0.00
 FUTURE VALUE            $0.00
ARROWS MOVE ->, * CALCULATES M RETURNS
```

Fig. 3-14. Annuity data entry form.

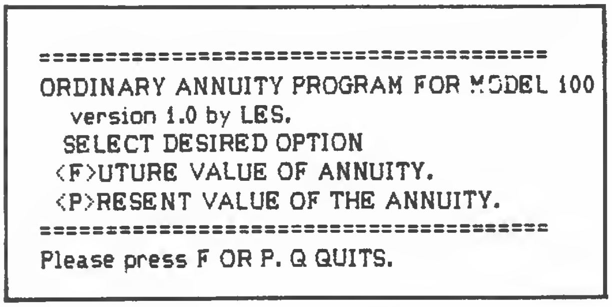
the item that you wish to calculate. If you want to quit or if you want to work a different type of annuity problem, press M and you will be returned to the first menu. You can quit by pressing Q or you can work another annuity problem by pressing F or P. You do not have to return to the first page of the menu if you want to work the same type of annuity problem with different data.

Let's work a few examples to show how the program works. (Note that all the examples use ANUIT2.BA. You can get the same results for ANUIT1.BA if you reduce the number of years by 1. That is, if the example says the life of the annuity is 5 years use 4 years to check out ANUIT1.BA.)

(Note that the programs assume that the compounding period for interest and the number of payments per year are the same. Before you run any of the examples, I suggest that you run the example given above to make sure that your program is debugged.)

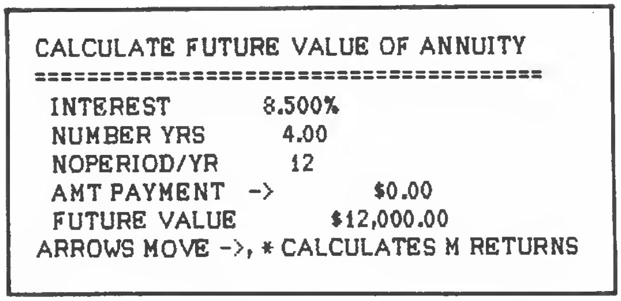
Example 7

You are interested in purchasing a new car four years from now. You estimate that the car will cost $12,000. You want to know how much you must save each month to have enough to purchase the car. You can earn 8.5 percent interest on your money. Run ANUIT2.BA. The display will clear and then you should see the first page of the data entry menu, Fig. 3-15. This is a future value problem, so press F. Now fill in the data entry form. You know everything except for the amount of the payment. Set it to 0. When you have entered all the data, the display should look like Fig. 8-16. Review them to be sure they are all correct. Then press * to do the calculations. When the calculations are complete, you should get Fig. 3-17.



```
======================================
ORDINARY ANNUITY PROGRAM FOR MODEL 100
   version 1.0 by LES.
  SELECT DESIRED OPTION
  <F>UTURE VALUE OF ANNUITY.
  <P>RESENT VALUE OF THE ANNUITY.
======================================
Please press F OR P. Q QUITS.
```

Fig. 3-15. Data entry menu for Example 7.

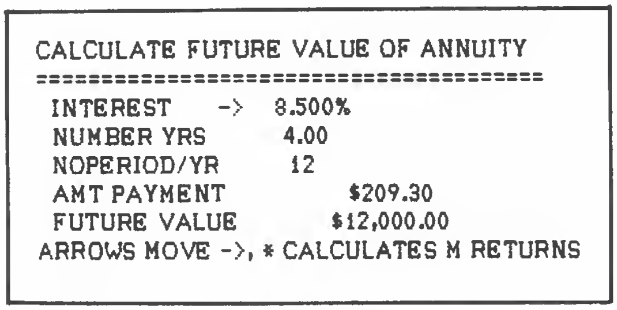


```
CALCULATE FUTURE VALUE OF ANNUITY
======================================
 INTEREST        8.500%
 NUMBER YRS       4.00
 NOPERIOD/YR     12
 AMT PAYMENT  ->           $0.00
 FUTURE VALUE          $12,000.00
ARROWS MOVE ->, * CALCULATES M RETURNS
```

Fig. 3-16. Completed data entry menu for Example 7.

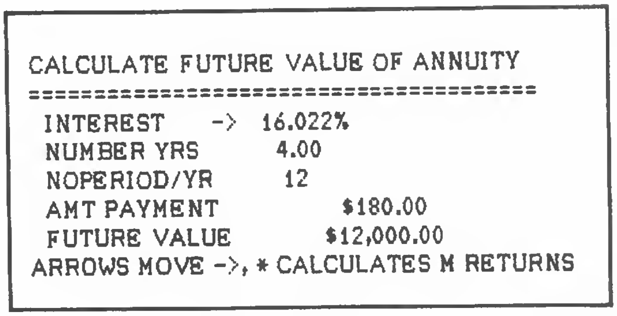
You're not too sure that you can find the necessary money in your budget. You can afford $250.00 a month. Use the program to calculate the interest rate necessary to end up with $12,000. Move the cursor to interest and enter 0. Then move the cursor to the AMT PAYMENT and enter $250. (Use the down arrow key to move the cursor.) When you have entered the data, press * to start the calculations again. Note that it takes some time because this is a trial-and-error computation. After a short while the calculations are completed and the answer is displayed as in Fig. 3-18. Note that the required interest rate is slightly over 16 percent.

You're not sure that you can really get 16 percent, so you wonder how much you can accumulate if you save the $250 a month and take the 8.5 percent interest the bank offers. The cursor already points to the interest so change it to 8.5. Use the down arrow to move the cursor to future value and enter 0. (You can use the up arrow and take advantage of the wrap-around feature of the data entry form.) You should have Fig. 3-19. Now press * and you should get Fig. 3-20.



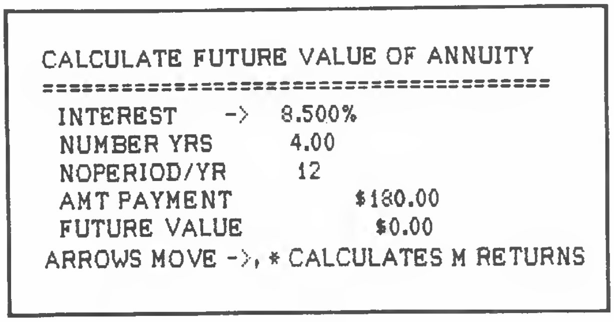
```
CALCULATE FUTURE VALUE OF ANNUITY
======================================
 INTEREST     ->  8.500%
 NUMBER YRS       4.00
 NOPERIOD/YR     12
 AMT PAYMENT             $209.30
 FUTURE VALUE          $12,000.00
ARROWS MOVE ->, * CALCULATES M RETURNS
```

Fig. 3-17. Calculated payment amount for Example 7.



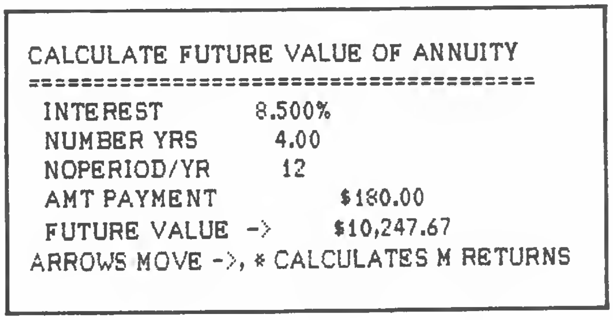
```
CALCULATE FUTURE VALUE OF ANNUITY
======================================
 INTEREST     ->  16.022%
 NUMBER YRS        4.00
 NOPERIOD/YR        12
 AMT PAYMENT               $180.00
 FUTURE VALUE            $12,000.00
ARROWS MOVE ->, * CALCULATES M RETURNS
```

Fig. 3-18. Recalculating Example 7 for interest rate.



```
CALCULATE FUTURE VALUE OF ANNUITY
======================================
 INTEREST     ->  8.500%
 NUMBER YRS        4.00
 NOPERIOD/YR        12
 AMT PAYMENT               $180.00
 FUTURE VALUE                $0.00
ARROWS MOVE ->, * CALCULATES M RETURNS
```

Fig. 3-19. Recalculating future value using 8.5% interest rate.

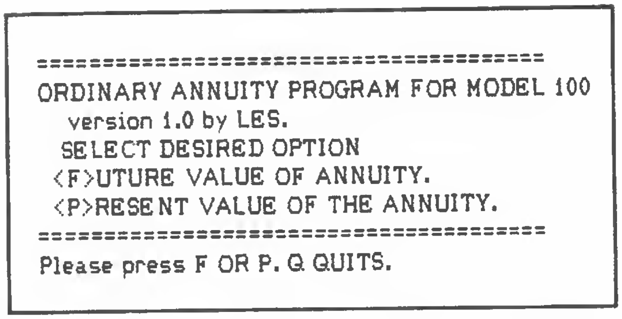


```
CALCULATE FUTURE VALUE OF ANNUITY
======================================
 INTEREST         8.500%
 NUMBER YRS        4.00
 NOPERIOD/YR        12
 AMT PAYMENT               $180.00
 FUTURE VALUE  ->        $10,247.67
ARROWS MOVE ->, * CALCULATES M RETURNS
```

Fig. 3-20. Completed calculations for Example 7.

This completes the session. Press M to return to the first page menu, and then press Q to quit and return to BASIC.

Example 8



```
======================================
ORDINARY ANNUITY PROGRAM FOR MODEL 100
   version 1.0 by LES.
  SELECT DESIRED OPTION
 <F>UTURE VALUE OF ANNUITY.
 <P>RESENT VALUE OF THE ANNUITY.
======================================
Please press F OR P. Q QUITS.
```

Fig. 3-21. Ordinary annuity data entry menu for Example 8.

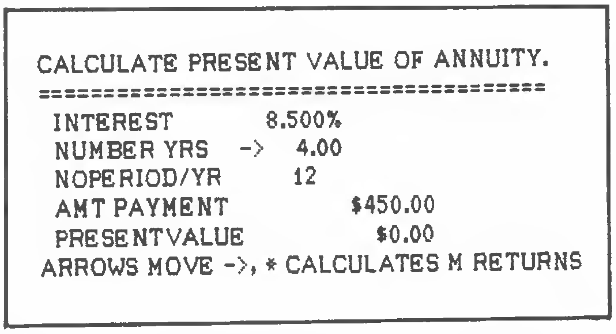
You are planning to send your child to college to learn to program TRS-80 computers. You have 10 years before the first payment is needed. You estimate that each month of the four years of college will cost $450.00. What kind of savings plan do you need to set up now to be sure that you have the money needed? You estimate that you can earn 8.5 percent interest on the money in the college account for the four years. You also estimate that you can earn 12.5 percent on the money you save between now and then.

This is a two-part problem. First you need to know how much you have to have on deposit to pay for four years of college. Then you need to know how much you need to save to come up with the required money.

The first part of this problem is a present value problem. You need to know how much money you need at the start of four years to pay at a rate of $450 per month, so run ANUIT2.BA.

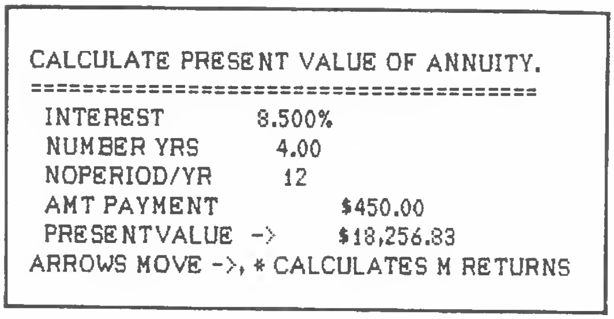
When you get the display Fig. 3-21 press P for present value. Then fill in the data entry form as shown in Fig. 3-22. Press * and get the answer as shown in Fig. 3-23. This is the amount that you must have to pay for four years of college.

Now let's answer the second part of the question. Press M to return to the first page of the menu. You want to work a future value problem now. (You want to have $18,256 saved in 10 years.) Press F to get Fig. 3-24. Note that all the data you previously entered are displayed. Move the cursor to IN-



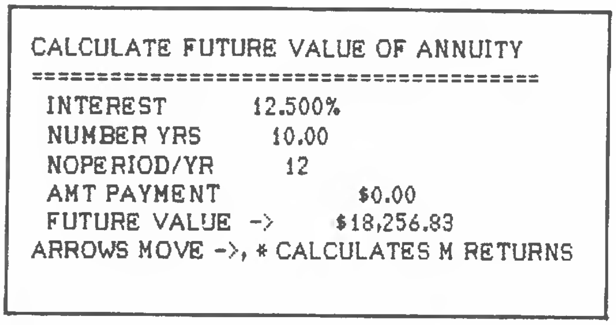
```
CALCULATE PRESENT VALUE OF ANNUITY.
======================================
 INTEREST         8.500%
 NUMBER YRS   ->   4.00
 NOPERIOD/YR        12
 AMT PAYMENT               $450.00
 PRESENTVALUE                $0.00
ARROWS MOVE ->, * CALCULATES M RETURNS
```

Fig. 3-22. Completed data entry form for Example 8.



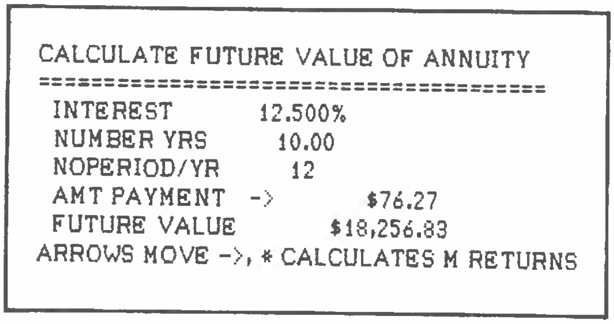
```
CALCULATE PRESENT VALUE OF ANNUITY.
=======================================
 INTEREST         8.500%
 NUMBER YRS        4.00
 NOPERIOD/YR       12
 AMT PAYMENT               $450.00
 PRESENTVALUE  ->          $18,256.83
ARROWS MOVE ->, * CALCULATES M RETURNS
```

Fig. 3-23. Present value calculation for Example 8.



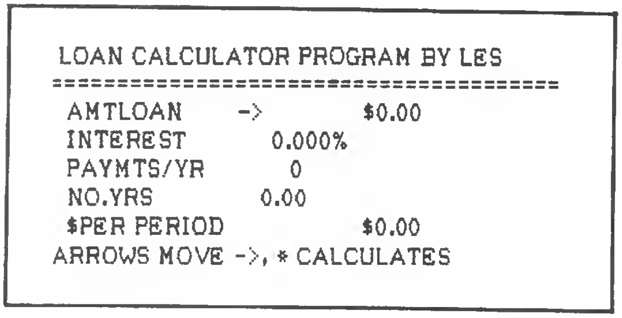
```
CALCULATE FUTURE VALUE OF ANNUITY
=======================================
 INTEREST         12.500%
 NUMBER YRS        10.00
 NOPERIOD/YR       12
 AMT PAYMENT                $0.00
 FUTURE VALUE  ->         $18,256.83
ARROWS MOVE ->, * CALCULATES M RETURNS
```

Fig. 3-24. Completed data entry form to calculate amount of payment.



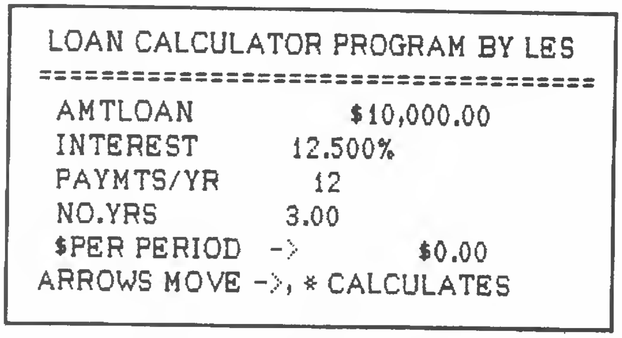
```
CALCULATE FUTURE VALUE OF ANNUITY
=======================================
 INTEREST         12.500%
 NUMBER YRS        10.00
 NOPERIOD/YR       12
 AMT PAYMENT  ->            $76.27
 FUTURE VALUE             $18,256.83
ARROWS MOVE ->, * CALCULATES M RETURNS
```

Fig. 3-25. Completed calculation for Example 8.



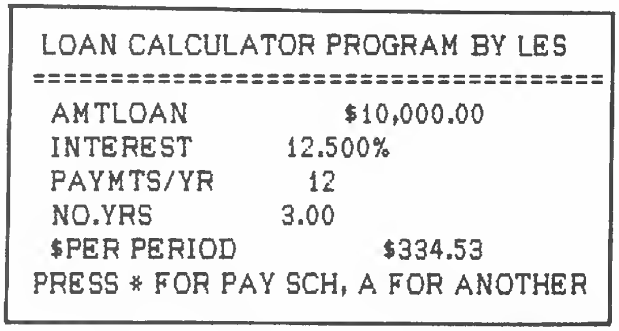
```
LOAN CALCULATOR PROGRAM BY LES
=======================================
 AMTLOAN      ->            $0.00
 INTEREST         0.000%
 PAYMTS/YR         0
 NO.YRS          0.00
 $PER PERIOD                $0.00
ARROWS MOVE ->, * CALCULATES
```

Fig. 3-26. Data entry form for loan calculation.



```
LOAN CALCULATOR PROGRAM BY LES
=======================================
 AMTLOAN                $10,000.00
 INTEREST         12.500%
 PAYMTS/YR          12
 NO.YRS           3.00
 $PER PERIOD  ->             $0.00
ARROWS MOVE ->, * CALCULATES
```

Fig. 3-27. Completed data entry for Example 9.



```
LOAN CALCULATOR PROGRAM BY LES
=======================================
 AMTLOAN                $10,000.00
 INTEREST         12.500%
 PAYMTS/YR          12
 NO.YRS           3.00
 $PER PERIOD                $334.53
PRESS * FOR PAY SCH, A FOR ANOTHER
```

Fig. 3-28. Monthly payment calculation for Example 9.

TEREST by pressing the down arrow and enter 12.5. The cursor automatically moved to NUMBER YRS; enter 10. You still have 12 periods per year, so move to AMT PAYMENT by pressing either the down arrow key or ENTER. Since you want to solve for the amount of the payment, enter 0 for AMT PAYMENT.

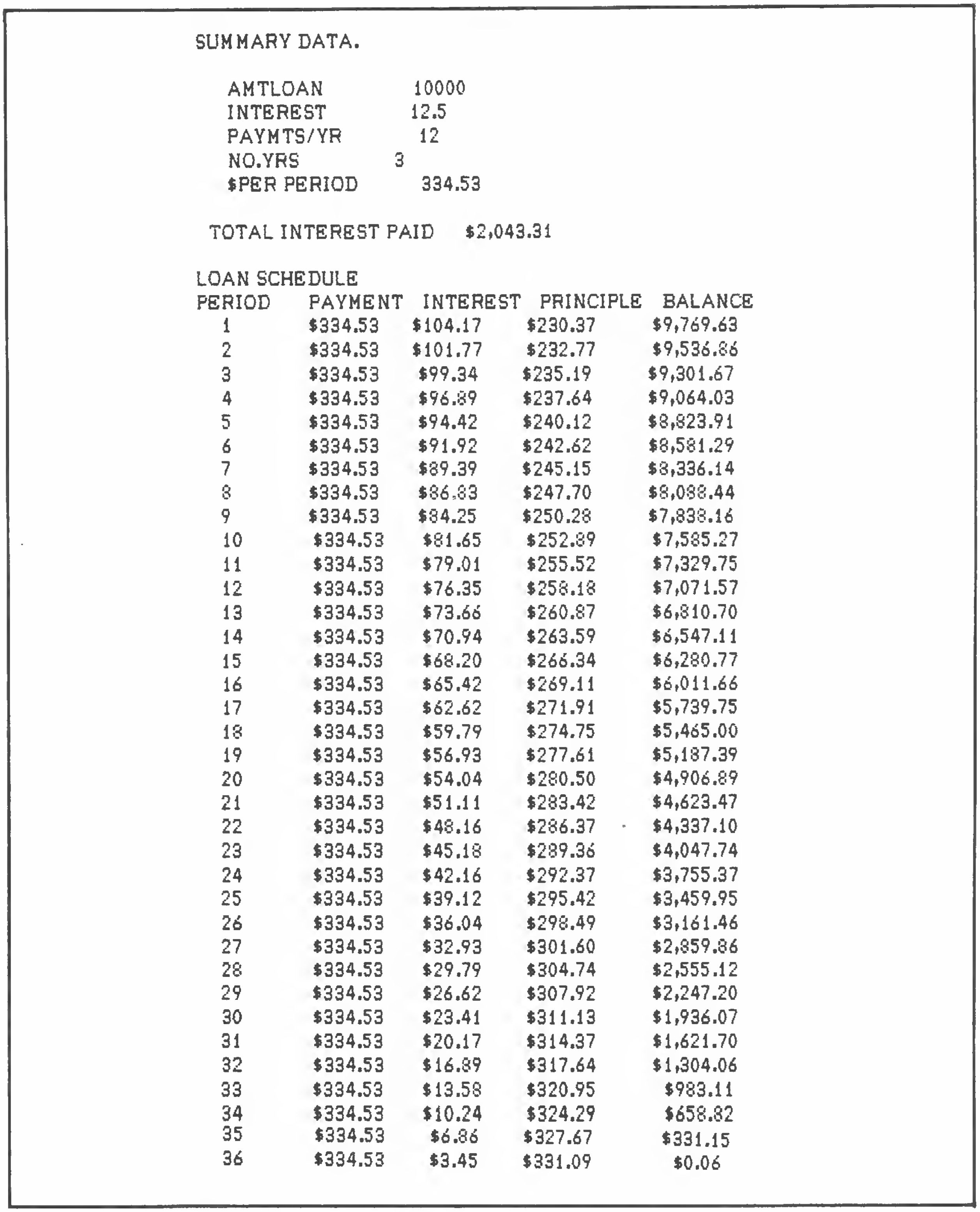
Press * and get the answer as displayed in Fig. 3-25. Now you are ready to set up the savings plan.

LOANS

Many loans are a form of annuity. Mortgages, car loans, and other loans where regular payments of principle and interest are made to pay off the loan are all annuities. The program LOAN.BA is designed to calculate the amount of the payment and, if you want, to print out a payment schedule that shows how much of each payment goes to interest and principal and the remaining balance. (This is the only program that provides explicit line printer output.) The data entry form is shown in Fig. 3-26.

Example 9—Loan Program

Suppose you want to borrow $10,000 for three



```
SUMMARY DATA.

   AMTLOAN          10000
   INTEREST         12.5
   PAYMTS/YR         12
   NO.YRS         3
   $PER PERIOD       334.53

 TOTAL INTEREST PAID    $2,043.31

LOAN SCHEDULE
PERIOD     PAYMENT   INTEREST   PRINCIPLE   BALANCE
  1        $334.53   $104.17    $230.37     $9,769.63
  2        $334.53   $101.77    $232.77     $9,536.86
  3        $334.53    $99.34    $235.19     $9,301.67
  4        $334.53    $96.89    $237.64     $9,064.03
  5        $334.53    $94.42    $240.12     $8,823.91
  6        $334.53    $91.92    $242.62     $8,581.29
  7        $334.53    $89.39    $245.15     $8,336.14
  8        $334.53    $86.83    $247.70     $8,088.44
  9        $334.53    $84.25    $250.28     $7,838.16
  10       $334.53    $81.65    $252.89     $7,585.27
  11       $334.53    $79.01    $255.52     $7,329.75
  12       $334.53    $76.35    $258.18     $7,071.57
  13       $334.53    $73.66    $260.87     $6,810.70
  14       $334.53    $70.94    $263.59     $6,547.11
  15       $334.53    $68.20    $266.34     $6,280.77
  16       $334.53    $65.42    $269.11     $6,011.66
  17       $334.53    $62.62    $271.91     $5,739.75
  18       $334.53    $59.79    $274.75     $5,465.00
  19       $334.53    $56.93    $277.61     $5,187.39
  20       $334.53    $54.04    $280.50     $4,906.89
  21       $334.53    $51.11    $283.42     $4,623.47
  22       $334.53    $48.16    $286.37     $4,337.10
  23       $334.53    $45.18    $289.36     $4,047.74
  24       $334.53    $42.16    $292.37     $3,755.37
  25       $334.53    $39.12    $295.42     $3,459.95
  26       $334.53    $36.04    $298.49     $3,161.46
  27       $334.53    $32.93    $301.60     $2,859.86
  28       $334.53    $29.79    $304.74     $2,555.12
  29       $334.53    $26.62    $307.92     $2,247.20
  30       $334.53    $23.41    $311.13     $1,936.07
  31       $334.53    $20.17    $314.37     $1,621.70
  32       $334.53    $16.89    $317.64     $1,304.06
  33       $334.53    $13.58    $320.95       $983.11
  34       $334.53    $10.24    $324.29       $658.82
  35       $334.53     $6.86    $327.67       $331.15
  36       $334.53     $3.45    $331.09         $0.06
```

Fig. 3-29. Hard copy of complete loan payment schedule from Example 9.

years to buy a new car. The interest rate is 12.5 percent and payments are due monthly. After filling in the form, you should see the screen display shown in Fig. 3-27. Calculating the monthly payment produces Fig. 3-28.

If you want a printout of the entire loan schedule, press *, and if you want to run another loan press A. In this case we want the loan schedule. The loan schedule is shown on the display one screen at a time. You change screens by pressing any key. At the end you can get a hard copy by pressing H. If you do not want the entire loan schedule displayed on the screen, you can press *. You are then asked if you want hard copy. So it is possible to get hard copy without having to go through several screens of payment schedule. The entire loan payment schedule for the example is given in Fig. 3-29.

Listing 3-1. Compound interest program COMPOUND.BA .

```
1 REM COMPOUND INTEREST PROGRAM
2 REM VERSION 1.0 BY LES FOR MODEL 100
3 REM 9/83
4 REM FOR TAB BOOKS
5 CLEAR
6 DEFINT J,L
11 A$="1234567890.+-"
12 AZ$=CHR$(13)+CHR$(31)+CHR$(30)+CHR$(42)+"M
      "+"m"
14 DIM F$(4),T$(5),X(5)
15 JR=16:
   JS=2:
   FORJ=0TO4:
   READF$(J):
   NEXTJ
16 FORJ=0TO4:
   READT$(J):
   NEXTJ
20 REM FOR TRS-80 MODEL 100 BY LES
40 J=1
45 NT=4
50 CLS:
   JF=0
60 PRINT STRING$(39,"=")
70 PRINT"COMPOUND INTEREST PROGRAM
72 PRINT "    version 1.0 by LES."
80 PRINT
85 :
   PRINTSTRING$(39,"=")
550 GOSUB 9000
560 REM FIND MISSING ITEM
570 J9=0
572 FORJ=0TONT:
   IFX(J)=0THEN J9=J9+1
```

```
574 NEXTJ
576 IFJ9>1THENCLS:
   PRINT" ERROR. TOO MANY UNKNOWNS.":
   FORJ=0TO500:
   NEXTJ:
   GOTO550
578 J=0
580 IFX(J)=0THEN600ELSEJ=J+1
590 IFJ>NTTHENCLS:
   PRINT"YOU SPECIFIED EVERYTHING. ERROR.":
   FORJ=1TO400:
   NEXTJ:
   GOTO550
595 GOTO 580
600 JS=J+2:
    LC=0:
    ONJGOTO630,640,660,680
605 PRINT@280,"   FINDING INTEREST RATE.";
610 C=X(4)/X(3):
   K=1/(X(1)*X(2)):
   R=C^K-1
620 R=R*100*X(2)
628 X(0)=INT(1000*R)/1000:
   GOTO550
630 PRINT@280,"CALCLATING NO. OF YRS.";:
   R=X(0)/100/X(2):
   R1=1+R:
   V=X(4)/X(3)
632 X(1)=LOG(V)/LOG(R1)/X(2)
638 GOTO 550
640 PRINT@280,"FINDING NR. OF PERIODS/YR";:
   LC=0:
   N=1:

641 V=X(4)/X(3):
   NL=1:
   NB=2000
642 R=X(0)/100/N:
   R1=1+R
643 XN=LOG(V)/LOG(R1)/X(1)
644 IFXN>2000THEN652
645 IF ABS(XN-N)/N<.000001THEN650
647 IFXN>NTHENNL=NELSENB=XN
648 N=(NL+NB)/2:
   LC=LC+1:
```

```
   IFLC>66THEN655 ELSE642
650 IF XN<1THEN 657 ELSE X(2)=INT(XN+.5):
   GOTO550LIST657
652 CLS:
   PRINT   ERROR ERROR >>>":
   PRINT"NOT POSSIBLE TO GET ANSWER.":
   PRINT"TRY NEW DATA.":
   PRINT:
   FORJ=0TO500:
   NEXTJ:
   GOTO550
655 CLS:
   PRINT"ERROR. ERROR."
656 PRINT "DID NOT CONVERGE. TRY NEW INPUT.":
   FORJ=0TO500:
   NEXTJ:
   GOTO550
657 CLS:
   PRINT"ERROR ERROR. ":
   PRINT"LESS THAN ONE PAYMENT/YR.":
   FORJ=0TO700:
   NEXTJ:
   GOTO550
660 PRINT@280,"CALCULATING PRESENT VALUE.";:
   R=X(0)/100/X(2)
665 X(3)=X(4)*(1+R)^-(X(2)*X(1))
670 GOTO 550
680 PRINT@280,"CALCULATING FUTURE VALUE.";:
   R=X(0)/100/X(2)
690 X(4)=X(3)*(1+R)^(X(2)*X(1))
695 GOTO 550
9000 REM DATA INPUT
9001 REM (C) 1983 L.E. SPARKS
9002 REM MAY BE FREELY USED
9010 CLS
9015 PRINT "COMPOUND INTEREST PROGRAM.
9020 PRINT STRING$(39,"=")
9040 FOR J=0TONT
9050    PRINTTAB(2) T$(J);TAB(21);:
   PRINTUSINGF$(J);X(J)
9060 NEXT J
9065 IF JQ=-9THEN9800
9070 PRINT @(JS*40)+JR,"->";
9075 PRINT@280,"ARROWS MOVE ->, * CALCULATES
     M RETURNS";
```

```
9080 Y$=INKEY$:
   IFY$=""THEN 9080
9090 IFINSTR(A$,Y$)<>0 THEN 9500
9095 IF Y$="M"ORY$="m"THEN JF=-9:
   RETURN
9100 JZ=INSTR(AZ$,Y$):
   IFJZ=0THEN9080
9110 ON JZ GOTO 9120,9120,9140,9250
9120 PRINT@(JS*40)+JR,"   ";
9122 JS=JS+1:
   IFJS>NT+2THEN JS=2
9124 PRINT@(JS*40)+JR,"->";
9130 GOTO 9080
9140 PRINT@(40*JS)+JR,"   ";
9150 JS =JS-1:
   IFJS<2 THEN JS=NT+2
9152 PRINT@(40*JS)+JR,"->";:
   GOTO9080
9250 RETURN
9500 PRINT @(40*JS)+JR,"          ";
9505 PRINT@280,"ENTER REST OF DATA PRESSENTER
      WHEN DONE";
9510 V$=Y$
9530 PRINT@(40*JS)+18,V$;CHR$(95);"   ";
9540 Y$=INKEY$:
   IF Y$ =""THEN 9540
9550 IF Y$=CHR$(13) THEN 9700
9560 IF Y$<>CHR$(29) THEN 9600
9570 IF LEN (V$)=0 THEN 9530
9580 V$=LEFT$(V$,LEN(V$)-1):

9590 Y$="":
   GOTO 9530
9600 IF INSTR(A$,Y$)=0 THEN 9530
9610 V$ = V$ + Y$
9620 GOTO 9530
9700 X(JS-2)=VAL(V$)
9710 JS=JS+1:
   IFJS>NT+2THENJS=2
9720 GOTO 9010
9800 JQ=0:
   PRINT@280,"PRESS * FOR PAY SCH, A FOR ANO
     THER";
9810 Y$=INKEY$:
   IFY$=""THEN9810
```

```
9815 IFY$="*"THEN9830ELSEIFY$="A"THEN90
34000 DATA ##.###%,##.##,###,"$$##,#######.##
     ","$$###,######.##"
35000 DATA INTEREST,NUMBER YRS,NOPERIOD/YR,PR
     ESVALUE,FUTVAL,END
```

Listing 3-2. Annuity due program ANUIT1.BA.

```
1 REM ANNUITY DUE PROGRAM VERSION 1.0
2 REM FOR TRS-80 MODEL 100 BY LES
3 REM ANUIT1.BA
5 CLEAR
6 DEFINT J,L
7 CL$=CHR$(27)+"E"
8 C1$=CHR$(27)+"K"
11 A$="1234567890.+-"
12 AZ$=CHR$(13)+CHR$(31)+CHR$(30)+CHR$(42)+"M"
     +"m"
14 DIM F$(4),T$(5),X(5)
15 JR=16:
   JS=2:
   FORJ=0TO4:
   READF$(J):
   NEXTJ
16 FORJ=0TO4:
   READT$(J):
   NEXTJ
40 J=1
50 CLS:
   JF=0
60 PRINT STRING$(39,"=")
70 PRINT "ANNUITY DUE PROGRAM FOR MODEL 100"
72 PRINT "    version 1.0 by LES."
80 PRINT"   SELECT DESIRED OPTION":
   PRINT"  <F>UTURE VALUE OF ANNUITY.":
   PRINT"  <P>RESENT VALUE OF THE ANNUITY."
85 :
   PRINTSTRING$(39,"=")
86 PRINT@280,"Please press F OR P. Q QUITS.";


90 Y$=INKEY$:
   IFY$=""THEN90ELSEJ=INSTR("QqFfPp",Y$)
92 IF J=0THEN90
95 IF J <3 THEN CLS:
    PRINT:
    FILES:
    END
100 IFJ>4THENGOSUB 900 ELSE GOSUB 500
110 GOTO 50
500 REM CALCULATE FUTURE VALUE
505 CLS
510 T$="CALCULATE FUTURE VALUE OF ANNUITY"
520 T$(4)="FUTURE VALUE"
540 NT=4
550 GOSUB 9000
552 IFY$="M"ORY$="m" THEN RETURN
555 PRINT@280,STRING$(39," ");
560 REM FIND MISSING ITEM
570 J9=0
572 FORJ=0TONT:
    IFX(J)=0THEN J9=J9+1
574 NEXTJ
576 IFJ9>1THENCLS:
    PRINT" ERROR. TOO MANY UNKNOWNS.":
    FORJ=0TO500:
    NEXTJ:
    GOTO550
578 J=0
580 IFX(J)=0THEN600ELSEJ=J+1
590 IFJ>NTTHENCLS:
    PRINT"YOU SPECIFIED EVERYTHING. ERROR.":
    FORJ=1TO400:
    NEXTJ:
    GOTO550
595 GOTO 580
600 JS=J+2:
     LC=0:
     ONJGOTO630,640,660,680
605 PRINT@280,"  FINDING INTEREST RATE.";
610 R=.15/X(2):
    RL=0:
    RB=1/X(2)
612 E1=(1+R)^(X(2)*X(1)):
    C1=X(3)*(E1-1)
615 R1=(1+R)*C1/X(4)
620 IF ABS((R-R1)/R)<.00001 THEN 628
624 IF LC>15THENCLS:
    PRINT" ERROR WILL NOT CONVERGE":
    PRINT"     TRY AGAIN":
```

```
   FORJ=0TO500:
   NEXTJ:
   GOTO550
625 LC=LC+1:
   IFR>R1THENRL=RELSERB=R
626 R=(RL+RB)/2:
   GOTO612
628 X(0)=R*100*X(2):
   X(0)=INT(1000*X(0))/1000:
   GOTO550
630 PRINT@280,"CALCLATING NO. OF YRS.";:
   R=X(0)/100/X(2):
   R1=1+R:
   V=X(4)/X(3)
632 C=V*R/R1+1:
   X(1)=LOG(C)/LOG(R1)/X(2)
638 GOTO 550
640 PRINT@280,"FINDING NR. OF PERIODS/YR";:
   LC=0:
   N=1:

642 R=X(0)/100/N:
   R1=1+R:
   V=X(4)/X(3):
   C=V*R/R1+1
644 XN=LOG(C)/LOG(R1)/X(1)
646 IF ABS(XN-N)/N<.001THEN650
648 N=XN:
   LC=LC+1:
   IFLC>16THEN655 ELSE642
650 IF XN<1THEN 657 ELSE X(2)=INT(XN+.5):
   GOTO550LIST657
655 CLS:
   PRINT"ERROR. ERROR."
656 PRINT "DID NOT CONVERGE. TRY NEW INPUT.":
   FORJ=0TO500:
   NEXTJ:
   GOTO550
657 CLS:
   PRINT"ERROR ERROR. ":
   PRINT"LESS THAN ONE PAYMENT/YR.":
   FORJ=0TO700:
   NEXTJ:
   GOTO550
660 PRINT@280,"CALCULATING AMT OF PAYMENT.";:
   R=X(0)/100/X(2)
665 X(3)=X(4)/(1+R)/((1+R)^(X(2)*X(1))-1)*R
670 GOTO 550
680 PRINT@280,"CALCULATING FUTURE VALUE.";:
   R=X(0)/100/X(2)
690 X(4)=X(3)*(1+R)*((1+R)^(X(2)*X(1))-1)/R
695 GOTO 550
900 CLS:
   T$="CALCULATE PRESENT VALUE OF ANNUITY."
910 T$(4)="PRESENTVALUE":
   NT=4:
   JS=2
920 GOSUB 9000
922 IFY$="M" OR Y$="m" THEN RETURN
925 PRINT@280,STRING$(39," ");
930 J9=0
940 FORJ=0TONT:
   IFX(J)=0THENJ9=J9+1
950 NEXTJ
960 IFJ9>1THENCLS:
   PRINT:
   PRINT"  ERROR ERROR.":
   PRINT"  TOO MANY UNKNOWNS.":
   FORJ=0TO500:
   NEXTJ:
   GOTO 920
970 J=0
980 IFX(J)=0THEN1010 ELSE J=J+1
990 IF J>NTTHENCLS:
   PRINT" ERROR. YOUR SPECIFIED EVERYTHING.":
   PRINT"TRY AGAIN.":
   FORJ=0TO500:
   NEXTJ:
   GOTO920
1000 GOTO 980
1010 JS=J+2:
   ON J GOTO 1100,1200,1300,1400
1019 PRINT@280,"FINDING INTEREST RATE.";
1020 REM FIND INTEREST RATE
1022 RL=0:
   RB=2.0/X(2):
   R=0.12'NOTE LIMITS MAX INTERESTRATE
1024 R1=.12/X(2):
   LC=0:
   C=X(4)/X(3)
```

```
1026 E1=(1-(1+R1)^-(X(1)*X(2)))/R1
1030 R=C/E1-1
1034 LC=LC+1
1036 IF ABS(R-R1)/R1<.001THEN1042
1038 IF LC>28THENCLS:
   PRINT"ERROR. ERROR.":
   PRINT"   DID NOT CONVERGE TRY NEW DATA.":
   FORJ=0TO500:
   NEXTJ:
   GOTO920
1039 IFR>R1THENRB=R1ELSERL=R1
1040 R1=(RL+RB)/2:
   GOTO1026
1042 X(0)=R1*100*X(2):
   GOTO920
1100 PRINT@280,"calculate no of years.";
1110 R=X(0)/100/X(2)
1120 C=X(4)/(X(3)*(1+R))*R
1122 IF C>=1THENCLS:
   PRINT"MONEY WILL LAST FOREVER @ THIS RATE.
     ":
   FORJ=0TO700:
   NEXTJ:
   X(1)=999999999:
   GOTO920
1130 X(1)=-1*LOG(1-C)/(X(2)*LOG(1+R))
1140 GOTO920
1200 PRINT@280," CALCULATE NO OF PERIODS/YR. "
     ;
1201 LC=0
1210 N1=6:
   R=X(0)/100
1212 R1=R/N1:
   C=X(4)/(X(3)*(1+R1))*R1
1214 IF C>=1THEN N1=N1*2:
   GOTO1212
1220 R1=R/N1:
   C=X(4)/(X(3)*(1+R1))*R1
1225 IF C>=1THENLC=LC+1:
   N1=N1+1:
   IFLC>22THEN1244ELSE1220
1230 N2=-1*LOG(1-C)/(X(1)*LOG(1+R1))
1235 LC=LC+1
1240 IF ABS(N2-N1)/N1<0.005THEN1250ELSE N1=(N2
     +N1)/2
```

```
1244 IFLC>20THENCLS:
   PRINT" ERROR ERROR":
   PRINT"DID NOT CONVERGE AFTER ";LC:
   FORJ=0TO600:
   NEXTJ:
   GOTO920
1245 GOTO 1220
1250 X(2)=INT(N2+.5):
   GOTO920
1300 PRINT@280,"CALCULATING AMOUNT OF PAYMENT.
     ";:
   R=X(0)/100/X(2)
1310 X(3)=X(4)*R/((1-(1+R)^-(X(2)*X(1))))/(1+R
     )
1320 GOTO 920
1400 PRINT@280,"CALCULATING PRESENT VALUE.";:
    R=X(0)/100/X(2):
   R1=(1+R)^-(X(1)*X(2))
1410 X(4)=X(3)/R*(1+R)*(1-R1)
1420 GOTO 920
9000 REM DATA INPUT
9001 REM (C) 1983 L.E. SPARKS
9002 REM MAY BE FREELY USED
9010 CLS
9015 PRINT T$
9020 PRINT STRING$(39,"=")
9040 FOR J=0TONT
9050   PRINTTAB(2) T$(J);TAB(21);:
   PRINTUSINGF$(J);X(J)
9060 NEXT J
9065 IF JQ=-9THEN9800
9070 PRINT @(JS*40)+JR,"->";
9075 PRINT@280,"ARROWS MOVE ->, * CALCULATES M
     RETURNS";
9080 Y$=INKEY$:
   IFY$=""THEN 9080
9090 IFINSTR(A$,Y$)<>0 THEN 9500
9095 IF Y$="M"ORY$="m"THEN JF=-9:
   RETURN
9100 JZ=INSTR(AZ$,Y$):
   IFJZ=0THEN9080
9110 ON JZ GOTO 9120,9120,9140,9250
9120 PRINT@(40*JS)+JR,"   ";
9122 JS=JS+1:
   IFJS>NT+2THEN JS=2
```

```
9124 PRINT@(JS*40)+JR,"->";
9130 GOTO 9080
9140 PRINT@(40*JS)+JR,"   ";
9150 JS =JS-1:
   IFJS<2 THEN JS=NT+2
9152 PRINT@(40*JS)+JR,"->";:
   GOTO9080
9250 RETURN
9500 PRINT@(40*JS)+JR,C1$;
9505 PRINT@280,"ENTER REST OF DATA PRESSENTER
      WHEN DONE";
9510 V$=Y$
9530 PRINT@(40*JS)+18,V$;CHR$(95);"   ";
9540 Y$=INKEY$:
   IF Y$ =""THEN 9540
9550 IF Y$=CHR$(13) THEN 9700
9560 IF Y$<>CHR$(29) THEN 9600
9570 IF LEN (V$)=0 THEN 9530
9580 V$=LEFT$(V$,LEN(V$)-1):

9590 Y$="":
   GOTO 9530
9600 IF INSTR(A$,Y$)=0 THEN 9530
9610 V$ = V$ + Y$
9620 GOTO 9530
9700 X(JS-2)=VAL(V$)
9710 JS=JS+1:
   IFJS>NT+2THENJS=2
9720 GOTO 9010
9800 JQ=0:
   PRINT@280,"PRESS * FOR PAY SCH, A FOR ANOT
     HER";
9810 Y$=INKEY$:
   IFY$=""THEN9810
9815 IFY$="*"THEN9830ELSEIFY$="A"THEN90
34000 DATA ##.###%,##.##,###,"$$##,#######.##"
     ,"$$###,######.##"
35000 DATA INTEREST,NUMBER YRS,NOPERIOD/YR,AMT
      PAYMENT,FUTVAL,END
```

Listing 3-3. Ordinary annuity program ANUIT2.BA.

```
1 REM ANNUITY PROGRAM VERSION 1.0
2 REM FOR TRS-80 MODEL 100 BY LES
3 REM ORDINARY ANNUITY VERSION
```

```
4 CLS
5 CLEAR
6 DEFINT J,L
7 CL$=CHR$(27)+"E"
8 C1$=CHR$(27)+"K"
11 A$="1234567890.+-"
12 AZ$=CHR$(13)+CHR$(31)+CHR$(30)+CHR$(42)+"M
     "+"m"
14 DIM F$(4),T$(5),X(5)
15 JR=16:
   JS=2:
   FORJ=0TO4:
   READF$(J):
   NEXTJ
16 FORJ=0TO4:
   READT$(J):
   NEXTJ
40 J=1
50 CLS:
   JF=0
60 PRINT STRING$(39,"=")
70 PRINT "ORDINARY ANNUITY PROGRAM FOR MODEL
     100"
72 PRINT "    version 1.0 by LES."
80 PRINT"   SELECT DESIRED OPTION":
   PRINT"  <F>UTURE VALUE OF ANNUITY.":
   PRINT"  <P>RESENT VALUE OF THE ANNUITY."


85 :
   PRINTSTRING$(39,"=")
86 PRINT@280,"Please press F OR P. Q QUITS.";

90 Y$=INKEY$:
   IFY$=""THEN90ELSEJ=INSTR("QqFfPp",Y$)
92 IF J=0THEN90
95 IF J <3 THEN CLS:
   PRINT:
   FILES:
   END
100 IFJ>4THENGOSUB 900 ELSE GOSUB 500
110 GOTO 50
500 REM CALCULATE FUTURE VALUE
505 CLS
510 T$="CALCULATE FUTURE VALUE OF ANNUITY"
```

```
520 T$(4)="FUTURE VALUE"
540 NT=4
550 GOSUB 9000
552 IFY$="M"ORY$="m" THEN RETURN
555 PRINT@280,STRING$(39," ");
560 REM FIND MISSING ITEM
570 J9=0
572 FORJ=0TONT:
   IFX(J)=0THEN J9=J9+1
574 NEXTJ
576 IFJ9>1THENCLS:
   PRINT" ERROR. TOO MANY UNKNOWNS.":
   FORJ=0TO500:
   NEXTJ:
   GOTO550
578 J=0
580 IFX(J)=0THEN600ELSEJ=J+1
590 IFJ>NTTHENCLS:
   PRINT"YOU SPECIFIED EVERYTHING. ERROR.":
   FORJ=1TO400:
   NEXTJ:
   GOTO550
595 GOTO 580
600 JS=J+2:
    LC=0:
    ONJGOTO630,640,660,680
605 PRINT@280,"  FINDING INTEREST RATE.";
610 R=.15/X(2):
   RL=0:
   RB=1/X(2)
612 E1=(1+R)^(X(2)*X(1)):
   C1=X(3)*(E1-1)
615 R1=C1/X(4)
620 IF ABS((R-R1)/R)<.00001 THEN 628
624 IF LC>15THENCLS:
   PRINT" ERROR WILL NOT CONVERGE":
   PRINT"     TRY AGAIN":
   FORJ=0TO500:
   NEXTJ:
   GOTO550
625 LC=LC+1:
   IFR>R1THENRL=RELSERB=R
626 R=(RL+RB)/2:
   GOTO612
628 X(0)=R*100*X(2):
   X(0)=INT(1000*X(0))/1000:
   GOTO550
630 PRINT@280,"CALCLATING NO. OF YRS.";:
   R=X(0)/100/X(2):
   R1=1+R:
   V=X(4)/X(3)
632 C=1+V*R:
   X(1)=LOG(C)/LOG(R1)/X(2)
638 GOTO 550
640 PRINT@280,"FINDING NR. OF PERIODS/YR";:
   LC=0:
   N=1:

642 R=X(0)/100/N:
   R1=1+R:
   V=X(4)/X(3):
   C=1+V*R
644 XN=LOG(C)/LOG(R1)/X(1)
646 IF ABS(XN-N)/N<.001THEN650
648 N=XN:
   LC=LC+1:
   IFLC>16THEN655 ELSE642
650 IF XN<1THEN 657 ELSE X(2)=INT(XN+.5):
   GOTO550LIST657
655 CLS:
   PRINT"ERROR. ERROR."
656 PRINT "DID NOT CONVERGE. TRY NEW INPUT.":
   FORJ=0TO500:
   NEXTJ:
   GOTO550
657 CLS:
   PRINT"ERROR ERROR. ":
   PRINT"LESS THAN ONE PAYMENT/YR.":
   FORJ=0TO700:
   NEXTJ:
   GOTO550
660 PRINT@280,"CALCULATING AMT OF PAYMENT.";:
   R=X(0)/100/X(2)
665 X(3)=X(4)/(1+R)/((1+R)^(X(2)*X(1))-1)*R
670 GOTO 550
680 PRINT@280,"CALCULATING FUTURE VALUE.";:
   R=X(0)/100/X(2)
690 X(4)=X(3)*((1+R)^(X(2)*X(1))-1)/R
695 GOTO 550
900 CLS:
    T$="CALCULATE PRESENT VALUE OF ANNUITY."
 910 T$(4)="PRESENTVALUE":
    NT=4:
    JS=2
 920 GOSUB 9000
 922 IFY$="M" OR Y$="m" THEN RETURN
 925 PRINT@280,STRING$(39," ");
 930 J9=0
 940 FORJ=0TONT:
    IFX(J)=0THENJ9=J9+1
 950 NEXTJ
 960 IFJ9>1THENCLS:
    PRINT:
    PRINT"  ERROR ERROR.":
    PRINT"  TOO MANY UNKNOWNS.":
    FORJ=0TO500:
    NEXTJ:
    GOTO 920
 970 J=0
 980 IFX(J)=0THEN1010 ELSE J=J+1
 990 IF J>NTTHENCLS:
    PRINT" ERROR. YOUR SPECIFIED EVERYTHING."
      :
    PRINT"TRY AGAIN.":
    FORJ=0TO500:
    NEXTJ:
    GOTO920
1000 GOTO 980
1010 JS=J+2:
    ON J GOTO 1100,1200,1300,1400
1019 PRINT@280,"FINDING INTEREST RATE.";
1020 REM FIND INTEREST RATE
1022 RL=0:
    RB=2.0/X(2):
    R=0.12'NOTE LIMITS MAX INTERESTRATE
1024 R1=.12/X(2):
    LC=0:
    C=X(3)/X(4)
1026 E1=(1-(1+R1)^-(X(1)*X(2)))
1030 R=C*E1
1034 LC=LC+1
1036 IF ABS(R-R1)/R1<.0001THEN1042
1038 IF LC>38THENCLS:
    PRINT"ERROR. ERROR.":
    PRINT"   DID NOT CONVERGE TRY NEW DATA.":
```

```
   FORJ=0TO500:
   NEXTJ:
   GOTO920
1039 R1=R
1040 GOTO1026
1042 X(0)=R1*100*X(2):
   GOTO920
1100 PRINT@280,"calculate no of years.";
1110 R=X(0)/100/X(2)
1120 C=X(4)/(X(3))*R
1122 IF C>=1THENCLS:
   PRINT"MONEY WILL LAST FOREVER @ THIS RATE
     .":
   FORJ=0TO700:
   NEXTJ:
   X(1)=999999999:
   GOTO920
1130 X(1)=-1*LOG(1-C)/(X(2)*LOG(1+R))
1140 GOTO920
1200 PRINT@280," CALCULATE NO OF PERIODS/YR.
     ";
1201 LC=0
1210 N1=6:
   R=X(0)/100
1212 R1=R/N1:
   C=X(4)/X(3)*R1
1214 IF C>=1THEN N1=N1*2:
   GOTO1212
1220 R1=R/N1:
   C=X(4)/(X(3)*(1+R1))*R1
1225 IF C>=1THENLC=LC+1:
   N1=N1+1:
   IFLC>22THEN1244ELSE1220
1230 N2=-1*LOG(1-C)/(X(1)*LOG(1+R1))
1235 LC=LC+1
1240 IF ABS(N2-N1)/N1<0.005THEN1250ELSE N1=(N
     2+N1)/2
1244 IFLC>20THENCLS:
   PRINT" ERROR ERROR":
   PRINT"DID NOT CONVERGE AFTER ";LC:
   FORJ=0TO600:
   NEXTJ:
   GOTO920
1245 GOTO 1220
1250 X(2)=INT(N2+.5):
   GOTO920
1300 PRINT@280,"CALCULATING AMOUNT OF PAYMENT
      .";:
   R=X(0)/100/X(2)
1310 X(3)=X(4)*R/((1-(1+R)^-(X(2)*X(1))))/(1+
     R)
1320 GOTO 920
1400 PRINT@280,"CALCULATING PRESENT VALUE.";:
    R=X(0)/100/X(2):
   R1=(1+R)^-(X(1)*X(2))
1410 X(4)=X(3)/R*(1-R1)
1420 GOTO 920
9000 REM DATA INPUT
9001 REM (C) 1983 L.E. SPARKS
9002 REM MAY BE FREELY USED
9010 CLS
9015 PRINT T$
9020 PRINT STRING$(39,"=")
9040 FOR J=0TONT
9050    PRINTTAB(2) T$(J);TAB(21);:
   PRINTUSINGF$(J);X(J)
9060 NEXT J
9065 IF JQ=-9THEN9800
9070 PRINT @(JS*40)+JR,"->";
9075 PRINT@280,"ARROWS MOVE ->, * CALCULATES
      M RETURNS";
9080 Y$=INKEY$:
   IFY$=""THEN 9080
9090 IFINSTR(A$,Y$)<>0 THEN 9500
9095 IF Y$="M"ORY$="m"THEN JF=-9:
   RETURN
9100 JZ=INSTR(AZ$,Y$):
   IFJZ=0THEN9080
9110 ON JZ GOTO 9120,9120,9140,9250
9120 PRINT@(40*JS)+JR,"   ";
9122 JS=JS+1:
   IFJS>NT+2THEN JS=2
9124 PRINT@(JS*40)+JR,"->";
9130 GOTO 9080
9140 PRINT@(40*JS)+JR,"   ";
9150 JS =JS-1:
   IFJS<2 THEN JS=NT+2
9152 PRINT@(40*JS)+JR,"->";:
   GOTO9080
9250 RETURN
```

```
9500 PRINT@(40*JS)+JR,C1$;
9505 PRINT@280,"ENTER REST OF DATA PRESSENTER
     WHEN DONE";
9510 V$=Y$
9530 PRINT@(40*JS)+18,V$;CHR$(95);"   ";
9540 Y$=INKEY$:
   IF Y$ =""THEN 9540
9550 IF Y$=CHR$(13) THEN 9700
9560 IF Y$<>CHR$(29) THEN 9600
9570 IF LEN (V$)=0 THEN 9530
9580 V$=LEFT$(V$,LEN(V$)-1):

9590 Y$="":
   GOTO 9530
9600 IF INSTR(A$,Y$)=0 THEN 9530
9610 V$ = V$ + Y$
9620 GOTO 9530
9700 X(JS-2)=VAL(V$)
9710 JS=JS+1:
   IFJS>NT+2THENJS=2
9720 GOTO 9010
9800 JQ=0:
   PRINT@280,"PRESS * FOR PAY SCH, A FOR ANO
     THER";
9810 Y$=INKEY$:
   IFY$=""THEN9810
9815 IFY$="*"THEN9830ELSEIFY$="A"THEN90
34000 DATA ##.###%,##.##,###,"$$##,#######.##
     ","$$###,######.##"
35000 DATA INTEREST,NUMBER YRS,NOPERIOD/YR,AM
     T PAYMENT,FUTVAL,END
```

Listing 3-4. Loan payment program LOAN.BA.

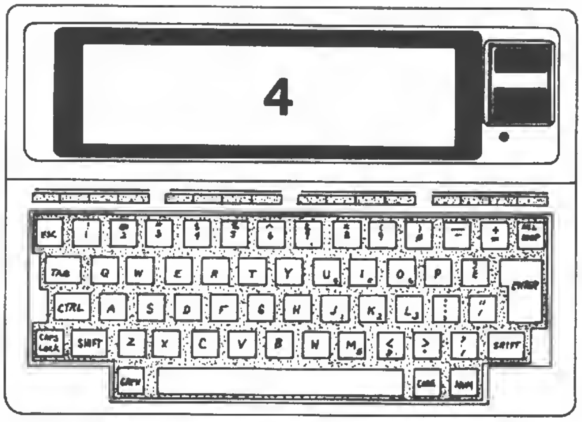
```
1 REM LOAN PAYMENT PROGRAM VERSION 1.1
2 REM 10/83 BY L. E. SPARKS
3 REM FOR TRS-80 MODEL 100
4 REM WRITTEN IN MICROSOFT BASIC
5 REM FOR TAB BOOKS
9 DEFINTJ,N,I:
   CLS
10 CLS
11 A$="1234567890.+-"
12 AZ$=CHR$(13)+CHR$(31)+CHR$(30)+CHR$(42)
14 PRINT STRING$(40," ,"):
   PRINT"LOAN PROGRAM BY LES."
15 JR=16
30 DIM T$(5),X(5),F$(5),P(90),PI(90),B(90)
35 FORJ=1TO5:
   READF$(J):
   NEXTJ
40 J=1
50 READT$:
   IFT$="END"THEN70ELSE T$(J)=T$
60 J=J+1:
   GOTO 50
70 NT=J-1
80 J1=2:
   JS=J1
90 GOSUB 9000
100 REM CHECK TO SEE WHAT IS MISSING
105 JC=0
110 FOR J=1TONT:
   IFX(J)=0THENJF=J:
   JC=JC+1
120 NEXTJ
130 IFJC=0THEN CLS:
   PRINT"ERROR YOU HAVE SPECIFIED ALL.":
   PRINT"REENTER DATA":
   FORJ=1TO1000:
   NEXT:
   GOTO90
140 IFJC>1THENCLS:
   PRINT"ERROR. TOO MANY UNKNOWNS.":
   PRINT"REENTER DATA.":
   FOR J=1TO1000:
   NEXTJ:
   GOTO90
150 ONJFGOTO200,300,400,500,700
200 R=X(2)/100/X(3)
210 R1=1-(1+R)^-(X(3)*X(4))
220 R2=R/R1:
   X(1)=INT(X(5)/R2*100)/100
230 GOTO90
300 R=12/X(3)/100
305 IFX(5)*X(3)*X(4)<X(1)THENR=0:
   GOTO312
310 R1=X(5)*(1-(1+R)^(-X(3)*X(4)))/X(1)
312 IFR>0THEN320ELSECLS:
   PRINT"CANNOT MAKE THIS LOAN."
```

```
314 PRINT"INTEREST RATE IS 0 OR LESS."
316 PRINT"PRESS ANYKEY TO CONTINUE."
318 IFINKEY$=""THEN318ELSE90
320 IF ABS(R-R1)/R<=.001 THEN 330 ELSER=R1
325 GOTO 310
330 X(2)=R*100*X(3)
340 GOTO 90
400 REM CALCULATE NO PERIODS/YR
410 N=1:
   JN=1
420 R=X(2)/100/N
430 R1=1-(1+R)^-(N*X(4))
440 XN=X(1)*X(2)/100/(X(5)*R1)
450 JN=JN+1:
   IFABS(XN-N)/N<=.1THEN480
460 N=INT(XN):
   IF JN=12THEN470ELSE420
470 CLS:
   PRINT:
   PRINT"DID NOT CONVERGE TO NO. PAYMENTS/YR"

472 PRINT"BEST GUESS IS ";INT(XN)
475 PRINT"PRESS ANY KEY TO CONTINUE"
476 Y$=INKEY$:
   IFY$=""THEN476ELSE90
480 X(3)=INT(XN+.5)
490 GOTO90
500 REM CALCULATE NO OF YEARS
510 R=X(2)/100/X(3)
520 AR=X(1)*R/X(5)
530 A1=1-AR
535 IFA1>0THEN540ELSECLS
536 PRINT"YOU CAN NOT MAKE SUCH A LOAN."
538 PRINT"PRESS ANYKEY TO CONTINUE."
539 INKEY$:
   IFINKEY$=""THEN539ELSE90
540 X(4)=-LOG(A1)/LOG(1+R)/X(3)
550 GOTO 90
700 REM CALCULATE MONTHLY PAYMENT
710 R=X(2)/100/X(3)
720 R2=1-(1+R)^-(X(3)*X(4))
730 R2=R2/R
740 X(5)=X(1)/R2
750 X(5)=INT(X(5)*100)/100 'ROUND TO CENT
760 JQ=-9
```

```
770 GOTO 90
9000 REM DATA INPUT
9001 REM (C) 1983 L.E. SPARKS
9002 REM MAY BE FREELY USED
9010 CLS
9015 PRINT" LOAN CALCULATOR PROGRAM BY LES"
9020 PRINT STRING$(39,"=")
9040 FOR J=1TONT
9050    PRINTTAB(2) T$(J);TAB(20);:
   PRINTUSINGF$(J);X(J)
9060 NEXT J
9065 IF JQ=-9THEN9800
9070 PRINT @(JS*40)+JR,"->";
9075 PRINT@280,"ARROWS MOVE ->, * CALCULATES";
9080 Y$=INKEY$:
   IFY$=""THEN 9080
9090 IFINSTR(A$,Y$)<>0 THEN 9500
9100 JZ=INSTR(AZ$,Y$):
   IFJZ=0THEN9080
9110 ON JZ GOTO 9120,9120,9140,9250
9120 PRINT@(JS*40)+JR,"    ";
9122 JS=JS+1:
   IFJS>NT+J1-1THEN JS=J1
9124 PRINT@(JS*40)+JR,"->";
9130 GOTO 9080
9140 PRINT@(40*JS)+JR,"    ";
9150 JS =JS-1:
   IFJS<J1 THEN JS=NT+J1-1
9152 PRINT@(40*JS)+JR,"->";:
   GOTO9080
9250 RETURN
9500 PRINT @(40*JS)+JR,"           ";
9505 PRINT@280,"ENTER REST OF DATA PRESSENTER
     WHEN DONE";
9510 V$=Y$
9530 PRINT@(40*JS)+18,V$;CHR$(95);"   ";
9540 Y$=INKEY$:
   IF Y$ =""THEN 9540
9550 IF Y$=CHR$(13) THEN 9700
9560 IF Y$<>CHR$(29) THEN 9600
9570 IF LEN (V$)=0 THEN 9530
9580 V$=LEFT$(V$,LEN(V$)-1):
9590 Y$="":
   GOTO 9530
9600 IF INSTR(A$,Y$)=0 THEN 9530
```

```
9610 V$ = V$ + Y$
9620 GOTO 9530
9700 X(JS-1)=VAL(V$)
9710 JS=JS+1:
   IFJS>NT+J1-1THENJS=J1
9720 GOTO 9010
9800 JQ=0:
   PRINT@280,"PRESS * FOR PAY SCH, A FOR ANOT
     HER";
9810 Y$=INKEY$:
   IFY$=""THEN9810
9815 IFY$="*"THEN9830ELSEIFY$="A"THEN90
9820 GOTO 9810
9830 REM FIGURE PAYMENT SCHEDULE
9840 REM P(I)=PRINCIPLE PAID/PER
9850 REM PI(I)=INTEREST PAID/PER
9860 REM B(I) = BALANCE LEFT PER PER.
9870 B=X(1):
   PT=0:
   PI=0:
    FOR J=1TO X(3)*X(4)
9880   PI(J)=R*B:
   P(J)=X(5)-INT(PI(J)*100)/100:
   PI=PI+PI(J):
   PT=PT+P(J):
   B(J)=B-P(J)
9885  B=B(J)
9890 NEXT J
9900 CLS 'NOW PRINT IT OUT
9910 PRINT"   LOAN SCHEDULE "
9920 PRINT@41,"PER   PAY$,  INT   PRIN  BA
     L"
9930 FOR J=1TOX(3)*X(4)
9935   PRINTUSING"###";J;
9940  PRINTX(5);TAB(12);INT(PI(J)*100)/100;TAB
     (19);P(J);TAB(27);
9960 PRINTUSING"##.##^^^^";B(J)
9970 IFINT(J/5)-J/5<>0THEN9990
9975 PRINT@280,"  PRESS ANY KEY TO CONTINUE *
     RETURNS";
9980 Y$=INKEY$:
   IFY$=""THEN9980
9982 IFY$="*"THEN 10000 ELSE CLS
9984 PRINT@41,"PER   PAY$,  INT   PRIN   BAL
     "
```

```
9990 NEXTJ
10000 PRINT@280,"PRESS H FOR HARDCOPY * RETURN
      S";
10010 Y$=INKEY$:
   IFY$=""THEN10010ELSEIFY$="*"THEN90ELSEIFY$
     <>"H"THEN10010
10020 CLS
10030 PRINT"  READY PRINTER:
   "
10040 LPRINT"   OUTPUT FROM LOAN PROGRAM."
10050 LPRINT
10060 LPRINT"SUMMARY DATA."
10070 LPRINT
10080 FOR J=1TONT:
   LPRINTTAB(5);T$(J);TAB(25);X(J):
   NEXTJ:
   LPRINT
10090 LPRINTUSING"  TOTAL INTEREST PAID $$#####
      ,###.##";PI
10110 LPRINT
10120 LPRINT"LOAN SCHEDULE"
10130 LPRINT"PERIOD";TAB(12)"PAYMENT";TAB(22)"
      INTEREST";TAB(33);"PRINCIPLE";TAB(45);"
      BALANCE"
10140 FOR J=1TOX(3)*X(4)
10150 LPRINTTAB(3);J;TAB(12);
10160 LPRINTUSING"$$####,###.##";X(5);:
   LPRINTTAB(25);
10170 LPRINTUSING"$$##,###.##";PI(J);:
   LPRINTTAB(38);:
   LPRINTUSING"$$####,###.##";P(J);:
   LPRINTTAB(55);
10180 LPRINTUSING"$$##,#####.##";B(J)
10190 NEXTJ
10200 GOTO 90
19999 DATA "$$####,#######.##",##.###%,###,###.
      ##,"$$####,#######.##"
20000 DATA AMTLOAN,INTEREST,PAYMTS/YR,NO.YRS,$
      PER PERIOD,END
```



Present Value and Financial Decisions

The fundamental decision rule of finance is this: *decisions should be selected to maximize the present value of the business.* This rule applies to both your personal as well as to your professional financial decisions.

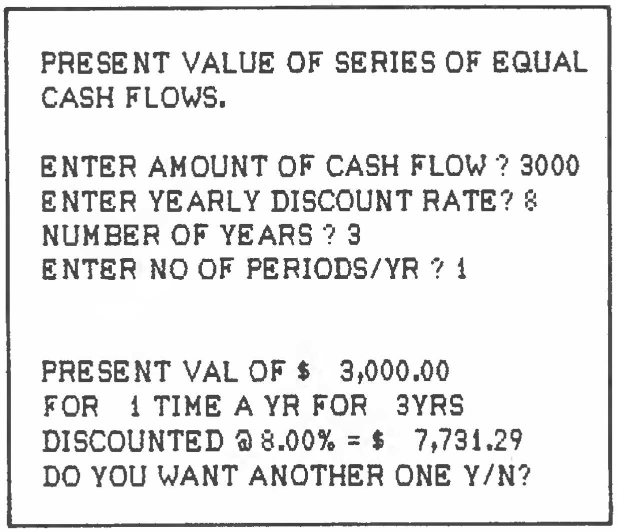
DISCOUNTING

The process of calculating the present value of a future amount is called *discounting.* The interest rate used to calculate the present value is called the *discount rate.* The power of the present value concept is that it allows you to compare cash flows from different time periods on a consistent basis. If you can handle compound interest problems and discounting problems, you can handle all the calculations necessary for financial decision making.

The discount rate has a large effect on the present value calculated for a given timing of cash flows. High discount rates favor near-term cash flows and thus tend to favor investments offering quick returns. Lower discount rates do not handicap the future returns. Thus the selection of the proper discount rate is of great importance.

The discount rate that should be selected for evaluation of an investment is the *cost of capital.* Exactly what is the cost of capital? There are two sources of capital—your own money and someone else's money. The cost of using someone else's money is the after-tax interest rate for using the money. The interest rate is stated in your loan agreement and is thus easy to determine. If all the capital used in an investment is borrowed, the cost of capital—and thus the discount rate—is fairly easy to determine.

But what about *your* money? There is no interest payment if you use your own funds, so you might think there is no cost associated with using your money. Not so! If you use your money for one investment, it is not available for other investments. For example, if you put your money in the stock market, you cannot put your money in a high interest certificate of deposit. The loss of the use of your money is called *opportunity cost.* The value of the opportunity cost is the return you can get on a



```
PRESENT VALUE OF SERIES OF EQUAL
CASH FLOWS.

ENTER AMOUNT OF CASH FLOW ? 3000
ENTER YEARLY DISCOUNT RATE? 8
NUMBER OF YEARS ? 3
ENTER NO OF PERIODS/YR ? 1


PRESENT VAL OF $  3,000.00
FOR  1 TIME A YR FOR  3YRS
DISCOUNTED @8.00% = $  7,731.29
DO YOU WANT ANOTHER ONE Y/N?
```

Fig. 4-1. Displays for example run of DISCNT.BA.

risk-free investment plus some extra risk premium. After all if a risky investment does not return more than a risk-free investment, why take the risk?

The concept of risk is important to investment and financial decisions. In general, the higher the risk the higher the expected return.

The program DISCNT.BA, Listing 4-1, can be used to calculate the present value of either a fixed future amount or the present value of a set of future cash flows. Let's look at an example using the program DISCNT.BA.

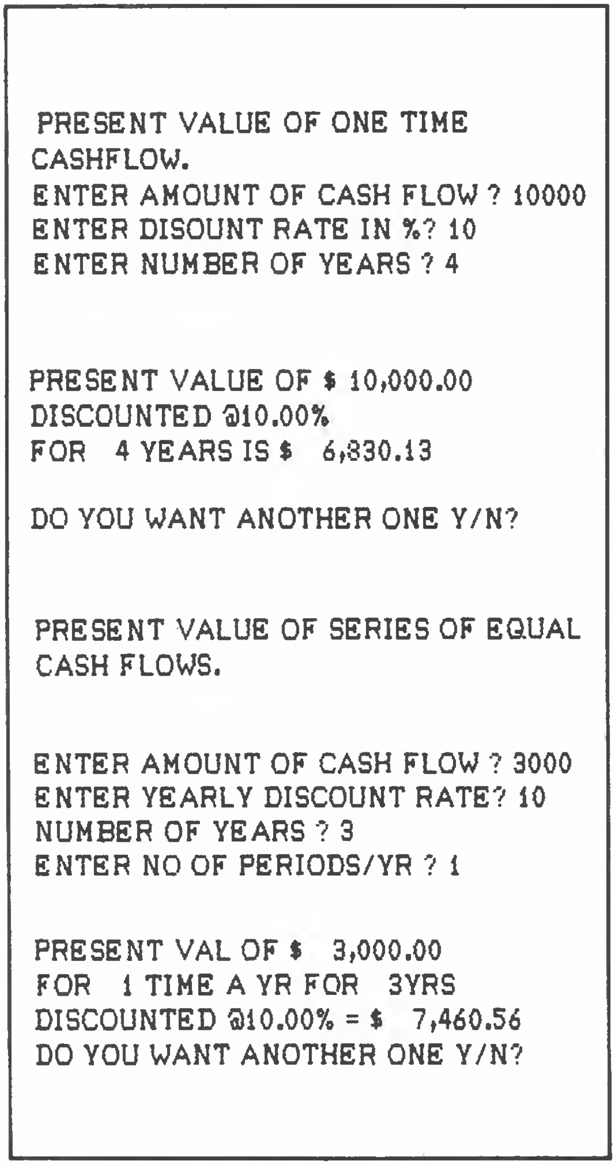
You are offered two investment opportunities. One will pay you $10,000 in a lump sum four years from now. The other investment will pay you $3,000 a year for three years. Both investments will cost $6,000, so which will you select? You will use your own money for the investment, and your best risk-free investment will return 7.5 percent. You judge the risk of this investment is not excessive, so you add a 0.5 percent risk premium to the risk-free return. The cost of capital for this investment is thus 8 percent. The run of DISCNT.BA for this example is given in Fig. 4-1.

Rerun the example with a higher discount rate to see the effect of discount rate on present value. Try 10 percent. The answer is shown in Fig. 4-2. Note that in this case the $3,000 per year is a better choice. These two runs demonstrate the principle that the high discount rates favor near term returns.

INCORPORATING PRESENT VALUE CONCEPTS

There are two general methods used to incorporate present value concepts into financial decisions. The first is the *net present value (NPV) method*, and the second is the *internal rate of return (IRR) method*. Of these two, the net present value method is the more common and is generally preferred for practical and theoretical reasons. Both techniques are discussed briefly below.

The net present value method directly calcu-



```
PRESENT VALUE OF ONE TIME
CASHFLOW.
ENTER AMOUNT OF CASH FLOW ? 10000
ENTER DISOUNT RATE IN %? 10
ENTER NUMBER OF YEARS ? 4


PRESENT VALUE OF $ 10,000.00
DISCOUNTED @10.00%
FOR  4 YEARS IS $  6,830.13

DO YOU WANT ANOTHER ONE Y/N?


PRESENT VALUE OF SERIES OF EQUAL
CASH FLOWS.


ENTER AMOUNT OF CASH FLOW ? 3000
ENTER YEARLY DISCOUNT RATE? 10
NUMBER OF YEARS ? 3
ENTER NO OF PERIODS/YR ? 1


PRESENT VAL OF $  3,000.00
FOR  1 TIME A YR FOR  3YRS
DISCOUNTED @10.00% = $  7,460.56
DO YOU WANT ANOTHER ONE Y/N?
```

Fig. 4-2. Displays for 10% discount rate.

lates the present value of the returns from an investment. The present value of the funds required to finance the investment is then subtracted from the present value of the returns. This difference is the ***net present value.*** If the net present value is positive, i.e., if the present value of the returns is greater than the present value of the equity investment, the investment is acceptable. If the net present value is negative, the investment is acceptable. If the net present value is negative, the investment is a poor one.

In situations where more than one investment is being analyzed (and you should always look at more than one situation) the investment giving the highest net present value is the preferred investment.

The major problem with the net present value method is finding the proper discount rate or interest rate to use in the calculations. In general, the discount rate should be equal to your cost of capital—whether the capital is borrowed or equity. The cost of borrowed is easy to determine. However, the cost of using equity (your own) funds is often difficult to determine. In fact, many people neglect the cost of their own money when they make financial decisions.

Your funds have a value because, if you did not have to use them for the investment, you could put them to work in a safe investment earning interest. You lose this interest when you make the investment.

The internal rate of return method was developed to overcome the problems of estimating a discount rate. The ***internal rate of return*** is that interest rate that gives a net present value of zero. That is, if the capital used to finance the investment were invested at the interest rate represented by the internal rate of return, you would end up with the same return as you got from the investment. The decision rule is to accept investments with internal rates of return greater than the cost of capital. (As you can see, we really only delayed the need to consider the cost of capital. It would obviously be a poor decision to accept investments that returned less than the cost of the funds needed to finance them.)

The internal rate of return method assumes that you can reinvest the cash flows from the investment at the IRR rate for the rest of the life of the business. This assumption can lead to problems in interpreting the results of internal rate of return analysis. This assumption can also result in two solutions for the internal rate of return—only one of which is correct.

In general, both the net present value and the internal rate of return methods will give the same ranking of investments and will lead to the same decision. However, there will be times when the two methods give very different answers. These times will generally be situations where the discount rate used in the net present value calculations is very different from the internal rate of return. The program provided in this section uses the net present value method.

The program here is designed with the limitations of the Model 100 in mind and thus produces less output than programs in other books. All the output produced by the program presented here is designed for the Model 100 display. No printer output is produced except, of course, for the printer output that can be produced by pressing the PRINT key on the Model 100.

The program is designed for situations where no residual value is attached to the investment and all the profits from the investment are expected to come from the cash flows generated by it, and for situations where much of the return is expected to come from the capital gains due to the investment.

Except for the last two items, "appreciation rate for property" and "base subject to appreciation," the items are self explanatory. The appreciation rate is the rate that you expect the property to increase in value. The ***base subject to appreciation*** is the amount of the investment that you expect to appreciate. For example, if you purchase property, part of your investment might be in furnishings that might not appreciate.

The program uses the new Internal Revenue Service (IRS) depreciation schedules to calculate the impact of depreciation on the investment. Note that the depreciation module is a separate part of the program and that it can be changed without too much trouble to incorporate any other depreciation scheme you want.

Program INVES1.BA

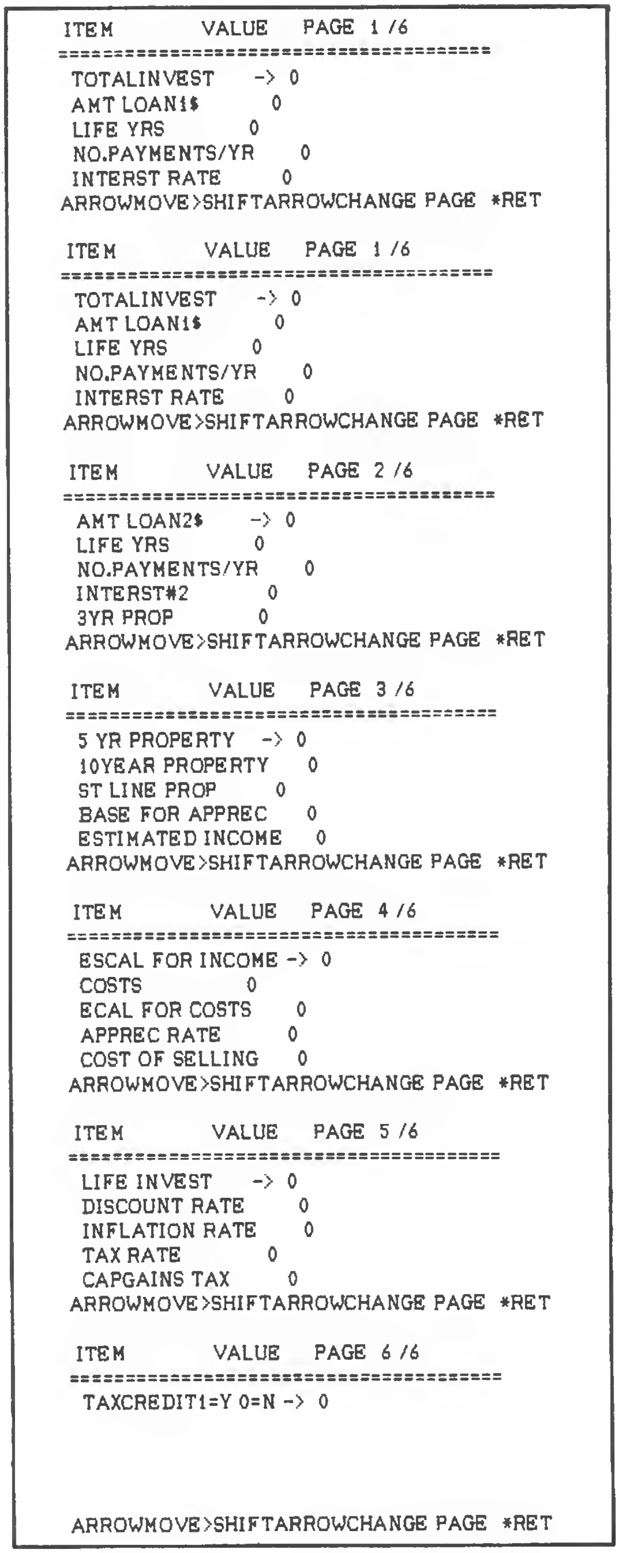
This program, shown in Listing 4-2 at the end of the chapter, is designed to calculate the net present value of an investment where the profit comes from the cash flows generated by the investment and also for the case where profit comes from resale of the investment. Examples of the first type are investments in new pieces of equipment, new product lines, or a new business. The program allows up to two loans for financing the investment.

Example of the second type of investment are purchases of real estate, stocks, and collectibles.

The input data requirements for the program are given in Table 4-1. For situations where expenses and/or income change with time, the program allows you to escalate the values by a constant percentage per year. The program uses the standard form for all data entry, as is shown in Fig. 4-3. Because there is more information than can be displayed on a single screen, the input data are divided into pages. You move from page to page with the shift up and down arrows. You use the unshifted up

Table 4-1. Data Needed for INVES1.BA Program.

Total Amount of Investment
Amount of First Loan
Interest Rate for First Loan
Life of First Loan
Number of Loan Payments per Year
Amount of Second Loan
Interest Rate for Second Loan
Life of Second Loan
Number of Loan Payments per Year Second Loan
Estimated Annual Income
Estimated Annual Expenses
Escalation for Income
Escalation for Expenses
Marginal Income Tax Rate
Discount Rate
Life of Investment
Value of 3 Year Property
Value of 5 Year Property
Value of 10 Year Property
Value of Straight Line Real Estate
Appreciation Rate for Property
Base Subject to Appreciation



```
 ITEM        VALUE   PAGE 1 /6
======================================
  TOTALINVEST    -> 0
  AMT LOAN1$       0
  LIFE YRS       0
  NO.PAYMENTS/YR     0
  INTERST RATE      0
ARROWMOVE>SHIFTARROWCHANGE PAGE  *RET

 ITEM        VALUE   PAGE 1 /6
======================================
  TOTALINVEST    -> 0
  AMT LOAN1$       0
  LIFE YRS       0
  NO.PAYMENTS/YR     0
  INTERST RATE      0
ARROWMOVE>SHIFTARROWCHANGE PAGE  *RET

 ITEM        VALUE   PAGE 2 /6
======================================
  AMT LOAN2$     -> 0
  LIFE YRS       0
  NO.PAYMENTS/YR     0
  INTERST#2       0
  3YR PROP       0
ARROWMOVE>SHIFTARROWCHANGE PAGE  *RET

 ITEM        VALUE   PAGE 3 /6
======================================
  5 YR PROPERTY   -> 0
  10YEAR PROPERTY     0
  ST LINE PROP      0
  BASE FOR APPREC     0
  ESTIMATED INCOME    0
ARROWMOVE>SHIFTARROWCHANGE PAGE  *RET

 ITEM        VALUE   PAGE 4 /6
======================================
  ESCAL FOR INCOME -> 0
  COSTS          0
  ECAL FOR COSTS     0
  APPREC RATE       0
  COST OF SELLING    0
ARROWMOVE>SHIFTARROWCHANGE PAGE  *RET

 ITEM        VALUE   PAGE 5 /6
======================================
  LIFE INVEST     -> 0
  DISCOUNT RATE      0
  INFLATION RATE      0
  TAX RATE        0
  CAPGAINS TAX      0
ARROWMOVE>SHIFTARROWCHANGE PAGE  *RET

 ITEM        VALUE   PAGE 6 /6
======================================
  TAXCREDIT1=Y 0=N -> 0



ARROWMOVE>SHIFTARROWCHANGE PAGE  *RET
```

Fig. 4-3. Display forms for INVES1.BA.

and down arrows to move from item to item on a single page. Calculations are initiated by pressing * in the same way as the other programs in this book. Calculated answers are displayed in tabular form similar to the input data.

Example 1

You are the financial manager for a small company. You are considering expanding into a new line of business. The people who would run the new business have produced the information on the new line shown in Table 4-2. They feel that this line of business will last about five years. Your company uses 12 percent as the cost of capital for new projects such as this one. Your job is to determine if the project is worth undertaking.

You use INVES1 to find the answer. Data entry for INVES1.BA uses the same format as the rest of the programs. Since there is far too much data to fit on one screen, the data are divided into pages. You move from page to page by pressing shift up and down arrow keys. The completed data entry screens are shown in Fig. 4-4.

When you are sure that the data are entered correctly, press * to start the calculations. The results of the calculations are shown in Fig. 4-5. Note that the program prints the summary of each year's business activity. You can examine the printout as long as you want. Press any key when you are ready to move on to the next year.

After you have looked at the last year, the computer prints out the summary for the investment. The key figure is, of course, the net present value. If this number is positive, the investment is acceptable. Note that for the example case the investment was a good one. If you are investigating several investments, you choose the investment that has the highest net present value. In cases where a single investment does not require all the capital available to invest, you should pick the combination of investments that gives the highest net present value.

INVES1.BA is also designed to cover situations where much of the return from the investment comes from capital gains. This means that the investment is sold at the end of the investment period. The total profit from the investment then becomes the profit from the sale plus operating profits over the life of the investment. One of the factors that influences the results of such an investment is the capital gains tax. The favorable treatment of capital gains often is the difference between good and bad investment.

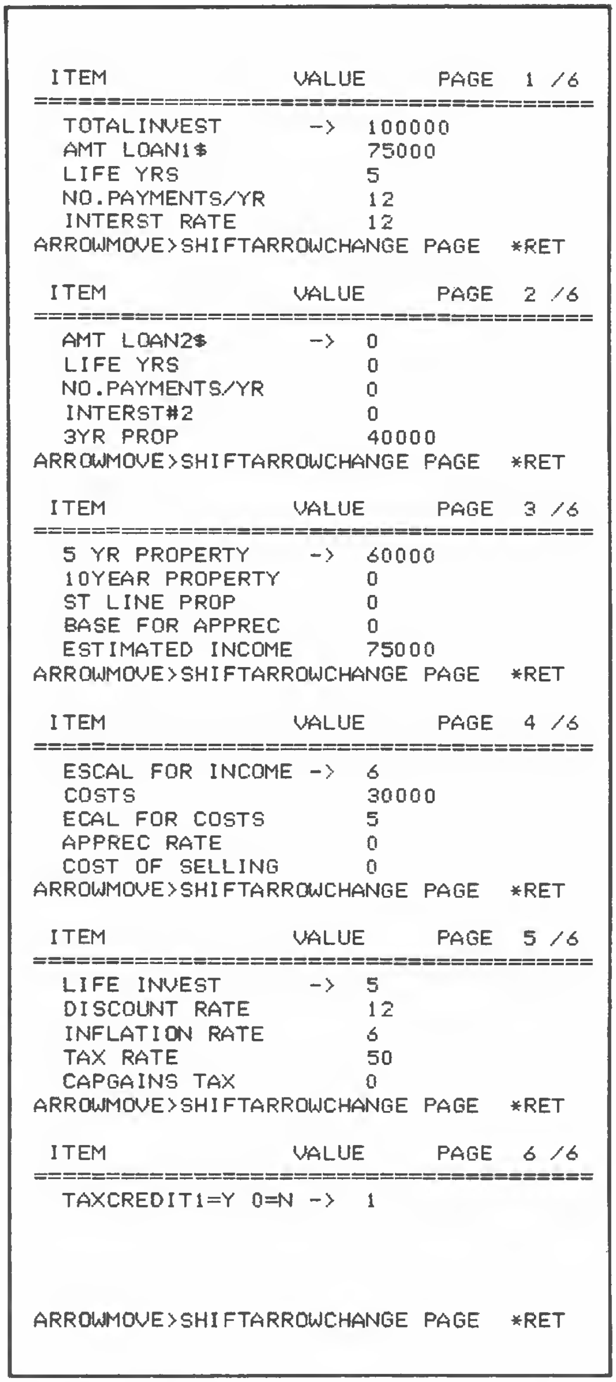
Table 4-2. Sample Data for Example 1 of INVES1.BA.

Total Amount of Investment	$100,000
Amount of First Loan	$ 75,000
Interest Rate for First Loan	12%
Life of First Loan	5 Years
Number of Loan Payments per Year	12
Amount of Second Loan	$0.00
Interest Rate for Second Loan	
Life of Second Loan	
No. Loan Payments per Yr 2nd Loan	
Estimated Annual Income	$75,000
Estimated Annual Expenses	$35,000
Escalation for Income	6%
Escalation for Expenses	5%
Marginal Income Tax Rate	50%
Discount Rate	12%
Inflation rate	6%
Life of Investment	5 years
Value of 3 Year Property	$40,000
Value of 5 Year Property	$60,000
Value of 10 Year Property	
Value of Straight Line Real Estate	

Example 2

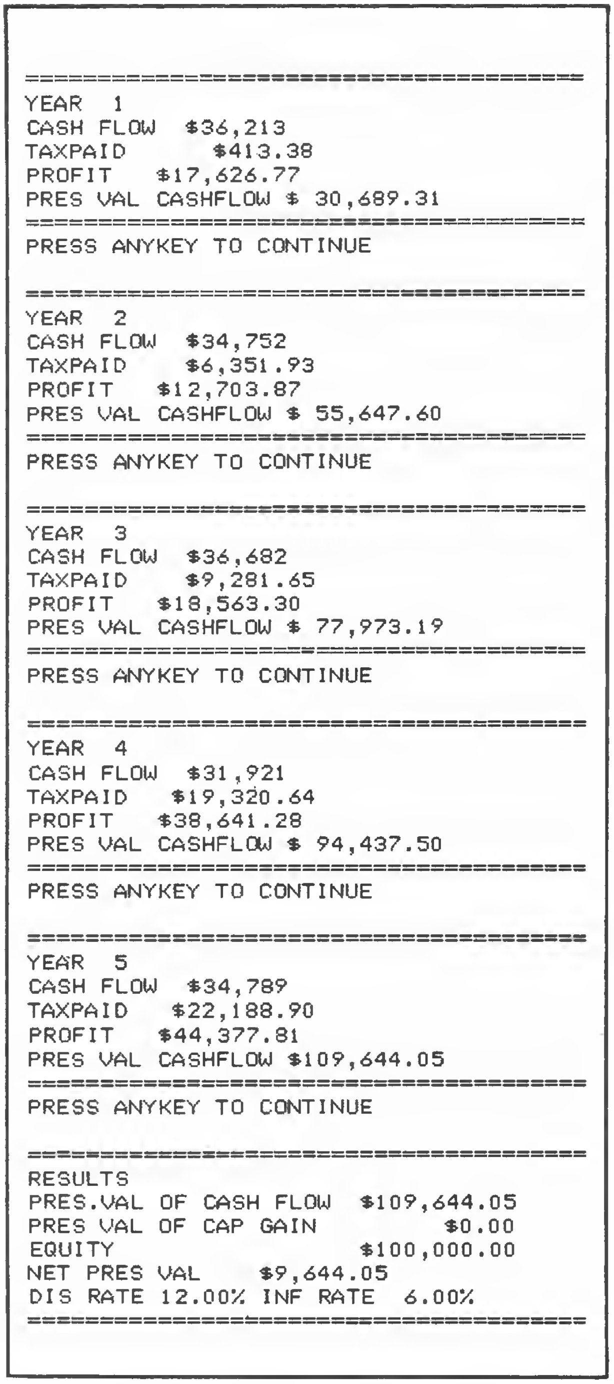
Suppose that the people offering the investment in the previous example offer to include facilities that can be used for the business. The facilities will add $50,000 to the cost of the investment. The advantage is that the facilities are expected to appreciate at 16 percent per year and can be sold at the end of five years. You also estimate that the new facilities will enable you to increase your estimated income with only a small increase in expense. The developers offer to loan you the additional $50,000 for three years at a favorable loan rate. Should you accept the deal? Use INVES1.BA to find out. Input data for Example 2 are shown in Table 4-3.

The data entry screens for this example are shown in Fig. 4-6, and the results are shown in the final screen display, Fig. 4-7. When the program is



```
 ITEM              VALUE     PAGE  1 /6
=======================================
  TOTALINVEST      ->  100000
  AMT LOAN1$           75000
  LIFE YRS             5
  NO.PAYMENTS/YR       12
  INTERST RATE         12
ARROWMOVE>SHIFTARROWCHANGE PAGE  *RET

 ITEM              VALUE     PAGE  2 /6
=======================================
  AMT LOAN2$       ->  0
  LIFE YRS             0
  NO.PAYMENTS/YR       0
  INTERST#2            0
  3YR PROP             40000
ARROWMOVE>SHIFTARROWCHANGE PAGE  *RET

 ITEM              VALUE     PAGE  3 /6
=======================================
  5 YR PROPERTY    ->  60000
  10YEAR PROPERTY      0
  ST LINE PROP         0
  BASE FOR APPREC      0
  ESTIMATED INCOME     75000
ARROWMOVE>SHIFTARROWCHANGE PAGE  *RET

 ITEM              VALUE     PAGE  4 /6
=======================================
  ESCAL FOR INCOME ->  6
  COSTS                30000
  ECAL FOR COSTS       5
  APPREC RATE          0
  COST OF SELLING      0
ARROWMOVE>SHIFTARROWCHANGE PAGE  *RET

 ITEM              VALUE     PAGE  5 /6
=======================================
  LIFE INVEST      ->  5
  DISCOUNT RATE        12
  INFLATION RATE       6
  TAX RATE             50
  CAPGAINS TAX         0
ARROWMOVE>SHIFTARROWCHANGE PAGE  *RET

 ITEM              VALUE     PAGE  6 /6
=======================================
  TAXCREDIT1=Y 0=N ->  1




ARROWMOVE>SHIFTARROWCHANGE PAGE  *RET
```

Fig. 4-4. Completed data entry screen for Example 1 of INVES1.BA.



```
=======================================
YEAR  1
CASH FLOW  $36,213
TAXPAID      $413.38
PROFIT   $17,626.77
PRES VAL CASHFLOW $ 30,689.31
=======================================
PRESS ANYKEY TO CONTINUE

=======================================
YEAR  2
CASH FLOW  $34,752
TAXPAID    $6,351.93
PROFIT   $12,703.87
PRES VAL CASHFLOW $ 55,647.60
=======================================
PRESS ANYKEY TO CONTINUE

=======================================
YEAR  3
CASH FLOW  $36,682
TAXPAID    $9,281.65
PROFIT   $18,563.30
PRES VAL CASHFLOW $ 77,973.19
=======================================
PRESS ANYKEY TO CONTINUE

=======================================
YEAR  4
CASH FLOW  $31,921
TAXPAID   $19,320.64
PROFIT   $38,641.28
PRES VAL CASHFLOW $ 94,437.50
=======================================
PRESS ANYKEY TO CONTINUE

=======================================
YEAR  5
CASH FLOW  $34,789
TAXPAID   $22,188.90
PROFIT   $44,377.81
PRES VAL CASHFLOW $109,644.05
=======================================
PRESS ANYKEY TO CONTINUE

=======================================
RESULTS
PRES.VAL OF CASH FLOW  $109,644.05
PRES VAL OF CAP GAIN         $0.00
EQUITY                 $100,000.00
NET PRES VAL     $9,644.05
DIS RATE 12.00% INF RATE  6.00%
=======================================
```

Fig. 4-5. Results of Example 1 using INVES1.BA.

used to study investments with capital gains, a summary of the selling action is also printed out. This summary tells how much the property sold for, how much was left after deducting the cost of selling, and how much was left after paying off any outstanding loans. Note that this investment was also favorable.

However, this second investment is not as

Table 4-3. Sample Data for Example 2 of INVES1.BA.

Total Amount of Investment	$150,000
Amount of First Loan	$ 75,000
Interest Rate for First Loan	12%
Life of First Loan	5 years
Number of Loan Payments per Year	12
Amount of Second Loan	$ 50,000
Interest Rate for Second Loan	10
Life of Second Loan	3 years
Number of Loan Payments per Year Second Loan	4
Estimated Annual Income	$100,000
Estimated Annual Expenses	$ 35,000
Escalation for Income	8%
Escalation for Expenses	10%
Base for Appreciation	$ 60,000
Appreciation Rate of Property	16%
Marginal Income Tax Rate	50%
Capital Gains Tax Rate	25%
Discount Rate	12%
Inflation Rate	6%
Life of Investment	5 years
Value of 3 Year Property	$ 40,000
Value of 5 Year Property	$ 50,000
Value of 10 Year Property	
Value of Straight Line Real Estate	$ 60,000

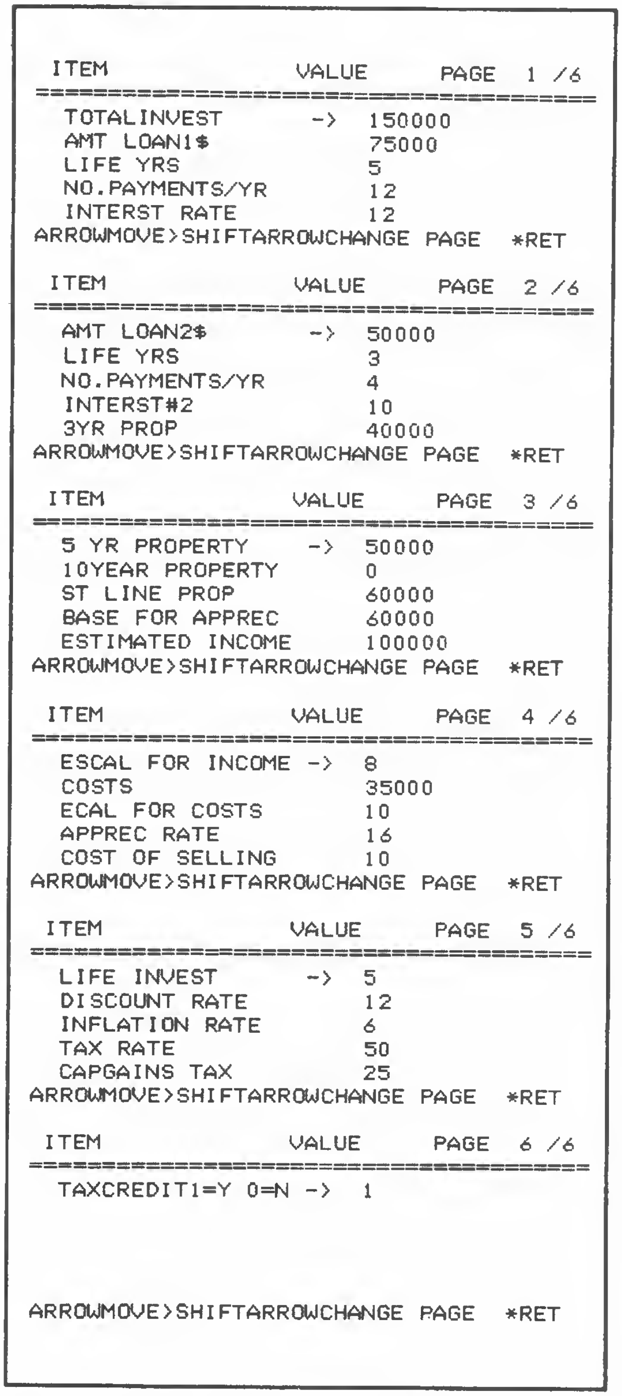
good as the first example. The net present value of the first investment was $9,644 and required an investment of only $100,000. This second investment returns a net present value of $6,574 on an investment of $150,000. Your recommendation should be to select the first investment.

Some Comments on INVES1.BA

The program INVES1.BA includes the effects of inflation on the investment by adding the inflation rate to the discount rate. The main disadvantage of handling inflation this way is that you might forget that inflation is included in the analysis. The final display shows the discount rate and the inflation rate to help remind you that inflation is included.

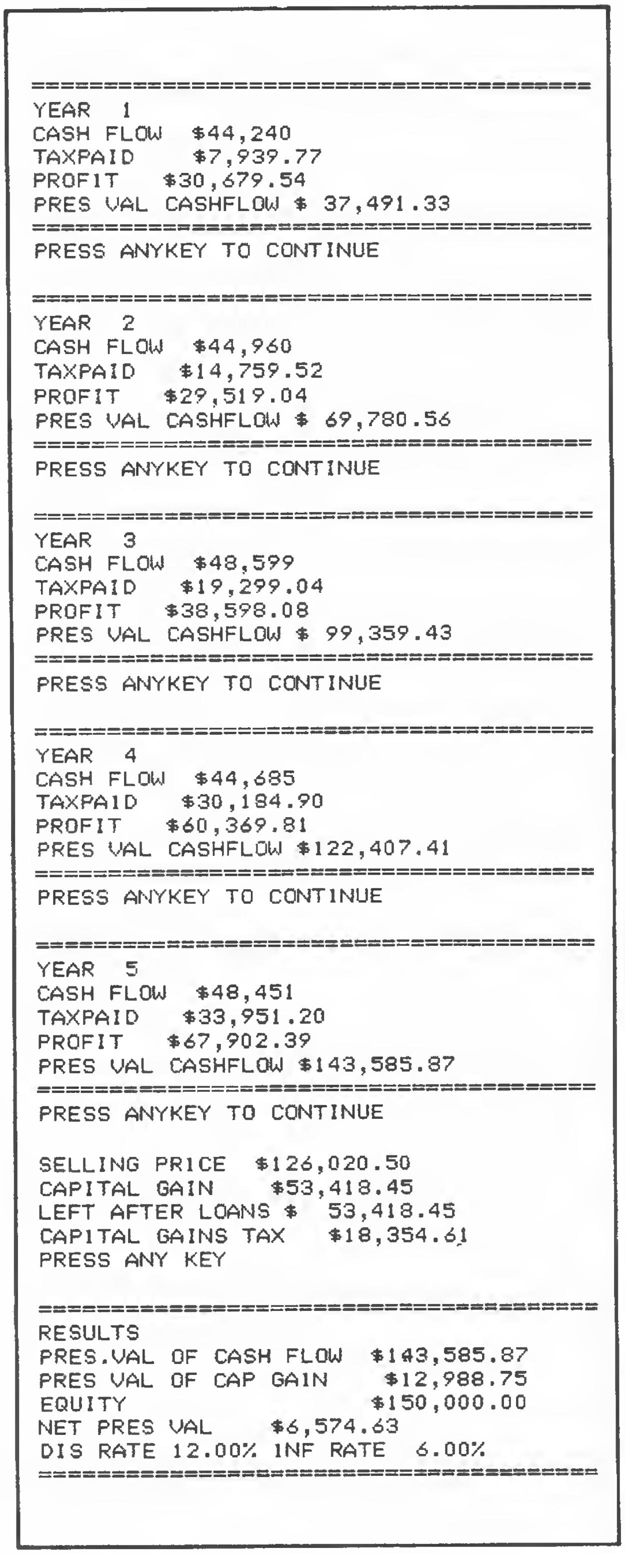
To help you get a better appreciation for the effects of inflation on your investments, I suggest that you analyze the investment first with a zero inflation rate and then with the expected inflation rate. See the next chapter for additional discussion of the effects of inflation.

This program allows consideration of the most common types of investments. However, the answer you get depends entirely on the data you entered. If the data are bad, then the answer is, too—even if the answer was produced by a com-



```
 ITEM               VALUE      PAGE  1 /6
=======================================
   TOTALINVEST      ->  150000
   AMT LOAN1$           75000
   LIFE YRS             5
   NO.PAYMENTS/YR       12
   INTERST RATE         12
ARROWMOVE>SHIFTARROWCHANGE PAGE   *RET

 ITEM               VALUE      PAGE  2 /6
=======================================
   AMT LOAN2$       ->  50000
   LIFE YRS             3
   NO.PAYMENTS/YR       4
   INTERST#2            10
   3YR PROP             40000
ARROWMOVE>SHIFTARROWCHANGE PAGE   *RET

 ITEM               VALUE      PAGE  3 /6
=======================================
   5 YR PROPERTY    ->  50000
   10YEAR PROPERTY      0
   ST LINE PROP         60000
   BASE FOR APPREC      60000
   ESTIMATED INCOME     100000
ARROWMOVE>SHIFTARROWCHANGE PAGE   *RET

 ITEM               VALUE      PAGE  4 /6
=======================================
   ESCAL FOR INCOME ->  8
   COSTS                35000
   ECAL FOR COSTS       10
   APPREC RATE          16
   COST OF SELLING      10
ARROWMOVE>SHIFTARROWCHANGE PAGE   *RET

 ITEM               VALUE      PAGE  5 /6
=======================================
   LIFE INVEST      ->  5
   DISCOUNT RATE        12
   INFLATION RATE       6
   TAX RATE             50
   CAPGAINS TAX         25
ARROWMOVE>SHIFTARROWCHANGE PAGE   *RET

 ITEM               VALUE      PAGE  6 /6
=======================================
   TAXCREDIT1=Y 0=N ->  1




ARROWMOVE>SHIFTARROWCHANGE PAGE   *RET
```

Fig. 4-6. Data entry screens for Example 2 of INVES1.BA.



```
=======================================
YEAR  1
CASH FLOW  $44,240
TAXPAID    $7,939.77
PROFIT   $30,679.54
PRES VAL CASHFLOW $ 37,491.33
=======================================
PRESS ANYKEY TO CONTINUE

=======================================
YEAR  2
CASH FLOW  $44,960
TAXPAID   $14,759.52
PROFIT   $29,519.04
PRES VAL CASHFLOW $ 69,780.56
=======================================
PRESS ANYKEY TO CONTINUE

=======================================
YEAR  3
CASH FLOW  $48,599
TAXPAID   $19,299.04
PROFIT   $38,598.08
PRES VAL CASHFLOW $ 99,359.43
=======================================
PRESS ANYKEY TO CONTINUE

=======================================
YEAR  4
CASH FLOW  $44,685
TAXPAID   $30,184.90
PROFIT   $60,369.81
PRES VAL CASHFLOW $122,407.41
=======================================
PRESS ANYKEY TO CONTINUE

=======================================
YEAR  5
CASH FLOW  $48,451
TAXPAID   $33,951.20
PROFIT   $67,902.39
PRES VAL CASHFLOW $143,585.87
=======================================
PRESS ANYKEY TO CONTINUE

SELLING PRICE  $126,020.50
CAPITAL GAIN    $53,418.45
LEFT AFTER LOANS $  53,418.45
CAPITAL GAINS TAX   $18,354.61
PRESS ANY KEY

=======================================
RESULTS
PRES.VAL OF CASH FLOW  $143,585.87
PRES VAL OF CAP GAIN     $12,988.75
EQUITY                 $150,000.00
NET PRES VAL    $6,574.63
DIS RATE 12.00% INF RATE  6.00%
=======================================
```

Fig. 4-7. Final screen display for Example 2.

puter. Spend some time making sure that the data are good.

An efficient way to use the program is to use rough estimates the first time through. If the investment is attractive, then spend some time refining your estimates and then rerun the program.

You should run the program several times with various values of the input data. By playing with the program in this way you get a feel for the investment not impossible with other methods of analysis. When you find an item to which the investment is very sensitive, spend a great deal of time making sure that you know the correct value of that item.

When you use the programs you should pay close attention to the profit and the cash flow printed in the year-by-year printout. If the cash flow is negative for very long, you may find you have serious problems with the investment. When the cash flow is negative, you will have to find additional funds to keep the investment going. If you can not find them, the investment might have to be liquidated at a loss. This can be especially serious in the case where much of the expected return is from out year capital gains.

EFFECTS OF INFLATION

Inflation makes financial decision making more difficult because it introduces uncertainty about the future value of the money returned from an investment. Thus one of the big impacts of inflation is that it increases the risk of financial decisions. This increase in risk must be compensated for in the form of increased potential return. But before we go any further in this discussion, we need to define terms.

Inflation is the increase in the general price level. When inflation exists, the purchasing power of future year dollars is less than the purchasing power of present year dollars. The inflation rate has a major effect on the profitability of an investment and it (the inflation rate) is generally poorly predicted. Inflation and its effect on the purchasing power of future year dollars should not be confused with the concept of present value.

Escalation is the rise in the price of a single good or service. The escalation rate for a given item may or may noτ resemble the general inflation rate. For example, computers have actually decreased in price (had negative escalation rates) at the same time the general inflation rate has been very high by historical standards. The interplay of inflation, supply and demand, and technology all determine the escalation rate for a given good or service.

Current year dollars are dollars received in a specific year. The purchasing power of current year dollars from different years may or may not be equal. The purchasing power of current year dollars depends on the inflation rate.

Constant year dollars are dollars adjusted to a constant purchasing power. Constant year dollars are used as an absolute ruler and provide the means of comparing cash flows from various years under conditions of inflation. The only way to increase your wealth is to increase the flow of constant year dollars.

The major effects of inflation on investments are:

1) Inflation effects the profitability of a project.
2) Inflation reduces the benefits of depreciation.
3) High rates of inflation increase the attractiveness of lease alternatives relative to capital investment.
4) Inflation favors projects that return cash as soon as possible.
5) High rates of inflation favor lower capital cost projects.
6) Inflation increases the risk of financial decisions.
7) Inflation makes future financial planning difficult.

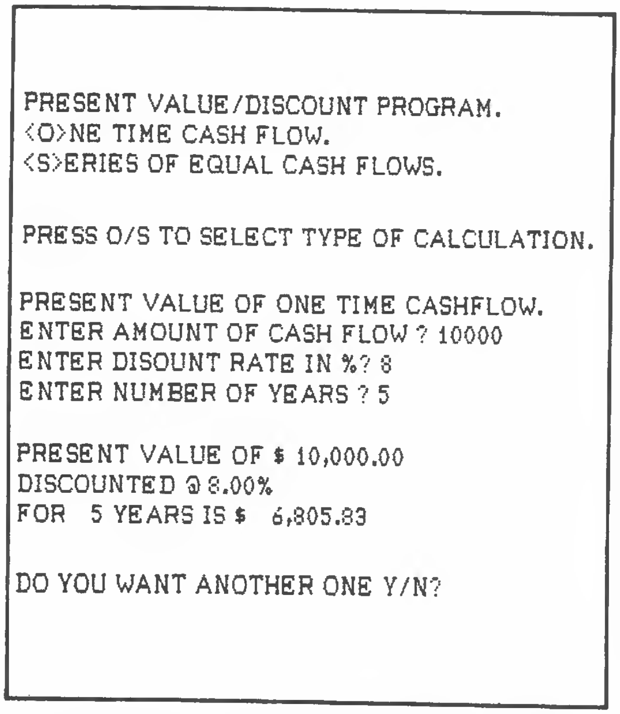
INFLATION AND FINANCIAL PLANNING

Inflation is generally incorporated into financial decision making by adding the expected inflation rate to the discount rate. Thus, if the cost of capital is 10 percent and the expected inflation rate is 8 percent, the discount rate used in calculating the present value of the returns from an investment is 18 percent. This is the approach used in the program INVES1.BA.

A second way to include inflation is to use the inflation rate to calculate the constant year dollar cash flow from an investment. The advantage of this method is that the inflation rate and the effects of inflation are clearly shown. There is no chance that the effects of inflation will be overlooked.

The discounting program presented earlier can be used to calculate the effects of inflation on a lump sum future payment. The program will tell you how much purchasing power, based on today's dollar, you can expect for a given future amount. You use the expected inflation rate for the discount rate. An example of this use of the program is shown below.

You expect to receive a lump sum of $10,000 five years from now. The expected inflation rate is 8 percent per year. How much will the $10,000 be worth in terms of today's dollars? We will use the DISCNT.BA to calculate the constant dollar value of the $10,000. Since this is a one-time cash flow problem, we select the O option from the menu. The results are shown in Fig. 4-8.



```
PRESENT VALUE/DISCOUNT PROGRAM.
<O>NE TIME CASH FLOW.
<S>ERIES OF EQUAL CASH FLOWS.

PRESS O/S TO SELECT TYPE OF CALCULATION.

PRESENT VALUE OF ONE TIME CASHFLOW.
ENTER AMOUNT OF CASH FLOW ? 10000
ENTER DISOUNT RATE IN %? 8
ENTER NUMBER OF YEARS ? 5

PRESENT VALUE OF $ 10,000.00
DISCOUNTED @ 8.00%
FOR   5 YEARS IS $   6,805.83

DO YOU WANT ANOTHER ONE Y/N?
```

Fig. 4-8. Using DISCNT.BA to estimate effects of inflation.

When you are analyzing a project and want to include the effects of inflation, you must be sure that all the figures you use include inflation effects in a consistent way. For example, be sure that the cost estimates and the expense estimates are consistent. Serious errors could occur if the cost figures did not include inflation effects and the income figures did.

Finally, try to use realistic inflation rates. Too high rates of inflation will lead to rejection of good projects. And of couse, too low rates of inflation will lead to acceptance of poor projects.

Forecasts of future inflation rates are regularly published by many groups and are generally reported in the financial press. You can also use the techniques presented in the chapter on forecasting to prepare your own forecasts. No matter how you come up with the inflation rate, you *must* include its effects in your financial decision making. Failure to do so will have unfortunate long-term consequences.

Listing 4-1. Discount program DISCNT.BA.

```
1 REM PROGRAM DISCNT.BA
10 REM DISCOUNT CALCULATION/PRESENT VALUE PRO
     GRAM
20 REM VERSION 1.0 SEP 83
30 REM BY LES FOR TRS-80 MODEL 100
40 REM ***************************
41 CLS
50 PRINT"PRESENT VALUE/DISCOUNT PROGRAM."
60 PRINT"<O>NE TIME CASH FLOW."
70 PRINT"<S>ERIES OF EQUAL CASH FLOWS."
80 PRINT
90 PRINT"PRESS O/S TO SELECT TYPE OF CALCULAT
     ION."
100 Y$=INKEY$:
   IFY$=""THEN100 ELSE IFINSTR("OoSs",Y$)=0T
     HEN100 ELSE IFINSTR("OoSs",Y$)<2 THEN
     GOSUB200 ELSE GOSUB300
110 PRINT"DO YOU WANT ANOTHER ONE Y/N?"
120 Y$=INKEY$:
   IFY$=""THEN120 ELSE IFY$="Y"THEN1 ELSE IF
     Y$="N"THEN END ELSE 120
200 CLS
210 PRINT"PRESENT VALUE OF ONE TIME CASHFLOW.
     "
220 INPUT"ENTER AMOUNT OF CASH FLOW ";C
230 INPUT"ENTER DISOUNT RATE IN %";R
240 INPUT"ENTER NUMBER OF YEARS ";N
250 P=C/(1+R/100)^N
260 CLS:
   PRINTUSING"PRESENT VALUE OF $#,#####.##";
     C:
   PRINTUSING"DISCOUNTED @##.##%";R
```

```
270 PRINT USING"FOR ### YEARS IS $#,######.##
     ";N,P
275 PRINT
280 RETURN
300 CLS
310 PRINT"PRESENT VALUE OF SERIES OF EQUAL"
320 PRINT"CASH FLOWS."
330 PRINT
340 INPUT"ENTER AMOUNT OF CASH FLOW ";C
350 INPUT"ENTER YEARLY DISCOUNT RATE";R
355 INPUT "NUMBER OF YEARS ";N
360 INPUT"ENTER NO OF PERIODS/YR ";M
372 R1=R/100/M
374 P=C*(1-(1/(1+R1)^(M*N)))/R1
385 CLS
390 PRINTUSING"PRESENT VAL OF $##,#####.##";C

391 IF N$<>"C"THEN392ELSEPRINTUSING"CONTINOUS
      DISCOUNTING FOR ### YRS";N:
   GOTO394
392 PRINTUSING"FOR ### TIME A YR FOR ###YRS";
     M,N
394 PRINTUSING"DISCOUNTED @##.##% = $#####,##
     .##";R,P
396 RETURN
```

Listing 4-2. Investment program INVES1.BA.

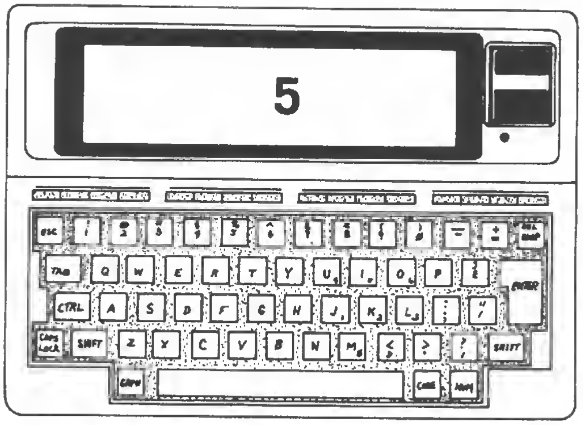
```
1 REM INVESTMENT ANALYSIS PROGRAM 1
2 REM VERSION 1.2 OCT83 BY L.E.SPARKS
7 REM FOR TAB BOOKS
8 DIM T$(30),X(30)
9 CLS:
   PRINTSTRING$(39,"="):
   PRINT:
   PRINT" INVESTMENT ANALYSIS 1 BY LES":
   PRINT" VERSION 1.1 FOR MODEL 100":
   PRINT" FOR TAB BOOKS":
   PRINTSTRING$(39,"="):
   FORJ=0TO20:
   GOSUB9900
51 A$="1234567890.+-"
52 AZ$=CHR$(13)+CHR$(31)+CHR$(30)+CHR$(42)+CHR
     $(2)+CHR$(20)
53 J1=1:
   J2 = 5
54 JS=2:
   JR=19:
   PG=1
60 GOSUB9000
61 PX=0:
   PZ=0
62 TX=0:
   PV=0:
   TL=0:
   GP=0
63 AT=0:
   P1=0
64 B1=0:
   B2=0:
   D3=0:
   D5=0:
   D9=0:
   DT=0:
   D8=0
65 REM HAVE DATA NOW DO CALCULATIONS
66 DR=1+X(22)/100+X(23)/100
67 AV=X(1)-X(2)-X(6)
70 REM FIRST LOAN PAYMENT CALCULATION
80 REM P1=PAYMENT,B1=BALANCE
85 B1=X(2):
   B2=X(6)
86 TC=.1*(.6*X(10)+X(11)+X(12))
87 IF X(26)=0THENTC=0
90 IFX(4)=0THENI1=0ELSEI1=X(5)/100/X(4)
91 IFX(8)=0THENI2=0ELSEI2=X(9)/100/X(8)
92 IFI1=0THEN93ELSEP1=X(2)*I1/(1-(1+I1)^(-X(4)
      *X(3)))
93 IFI2=0THEN95ELSEP2=X(6)*I2/(1-(1+I2)^(-X(7)
      *X(8)))
95 CLS:
   PRINT"BEGIN CALCULATIONS"
96 PV=0
100 J=1
105 A1=0:
   A2=0
110 PRINT"YEAR = ";J ;" OF ";X(21)
120 REM INCOME AND EXPENSE CALC
130 E1=X(15)*(1+X(16)/100)^(J-1)
132 EX=X(17)*(1+X(18)/100)^(J-1)
135 GP=E1-EX
136 IF J>X(3)THENIFJ>X(7)THEN145
```

```
137 A1=0:
   A2=0
140 GOSUB1000'INTEREST CALCULATIONS
141 AT=AT+A1+A2'TOTAL EQUITY
145 GOSUB2000'CALCULATE DEPRECIATION
150 GP=GP-T1-T2:
   G2=GP-DT
160 IFG2<=0THEN178'NO PROFIT NO TAX
161 TX=G2*X(24)/100:
   IFTL>TXTHENTX=0:
   TL=TL-TX:
   G3=G2:
   GOTO180
163 TX=TX-TL
164 IF TC=0THEN175
172 IFTX<TCTHENTX=0:
   TC=TC-TX:
   GOTO175
173 TX=TX-TC:
   TC=0
175 G3=G2-TX:

177 GOTO180
178 TL=ABS(TL)+ABS(G2)
179 G3=G2
180 CLS:
   PRINTSTRING$(39,"="):
   CF=G3+DT:
   PRINT"YEAR ";J:
   PRINTUSING"CASH FLOW$$###,###";CF
181 PV=PV+CF*(DR)^-J
182 PRINTUSING"TAXPAID $$###,###.##";TX
183 PRINTUSING"PROFIT $$###,###.##";G2
184 PRINTUSING"PRES VAL CASHFLOW $#####,#.##";
     PV
185 PRINTSTRING$(39,"=")
186 PRINT"PRESS ANYKEY TO CONTINUE";
187 Y$=INKEY$:
   IFY$=""THEN187
300 J=J+1
310 IFJ>X(21)THEN320ELSECLS:
   GOTO110
320 CLS:
   IFX(19)=0THEN410ELSE SP=X(14)*(1+X(19)/100
   )^X(21):
```

```
  PRINTUSING"SELLING PRICE $$######,#.##";SP


325 CS=SP*X(20)/100:
   SP=SP-CS
330 CG=SP-X(14):
   PRINTUSING"CAPITAL GAIN $$######,##.##";CG


340 CP=CG-B1-B2:
   PRINTUSING"LEFT AFTER LOANS $#####,###.##";
     CP
344 IF X(21)>15THENCG=CG+X(13):
   GOTO350
345 CG=CG+X(13)/15*X(21)'TAXABLE CAPITAL GAIN
     ADJUSTED FOR DEPRECIATION
350 CT=CG*X(25)/100:
   PZ=CP-CT:
   PX=PZ*DR^-J
355 PRINTUSING"CAPITAL GAINS TAX  $$#####,#.##"
     ;CT
360 PRINT"PRESS ANY KEY":
   Y$=INPUT$(1)
410 CLS:
   PRINTSTRING$(39,"="):
   PRINT"RESULTS"
420 PRINTUSING"PRES.VAL OF CASH FLOW $$#####,##
     .##";PV
430 PRINTUSING"PRES VAL OF CAP GAIN $$###,####
     .##";PX
435 PRINTUSING"EQUITY              $$###,####
     .##";AT+(X(1)-X(2)-X(6))'TOTAL EQUITY N
     OTE NOT DISCOUNTED
445 PRINTUSING"NET PRES VAL $$###,###.##";PV+P
     X-X(1)
446 PRINTUSING"DIS RATE ##.##% INF RATE ##.##%
     ";X(22);X(23)
450 PRINTSTRING$(39,"=");
460 Y$=INKEY$:
   IFY$=""THEN460ELSE60
1000 IFJ>X(2)THEN1040ELSEIFP1=0ORB1=0THEN1040


1005 IF B1<=1E-6THENB1=0:
   A1=0
1010 T1=0:
   A1=0:
```

```
1020 FORJJ=1TOX(4):
   P=B1*I1:
   PP=P1-P:
   B1=B1-PP:
   T1=T1+P:
   A1=A1+PP:
   NEXTJJ
1030 IFB1<0THENB1=0
1040 IFP2=0ORB2<=1E-5THENRETURN
1050 T2=0:
   A2=0
1060 FORJJ=1TOX(8):
   P=B2*I2:
   PP=P2-P:
   B2=B2-PP:
   T2=T2+P:
   A2=A2+PP:
   NEXTJJ
1070 IFB2<0THENB2=0
1080 RETURN
2000 REM DEPRECIATION CALCULATIONS
2010 REM YOU MAY REPLACE IF DESIRED
2020 REM USES ACCELERATED RECOVERY
2025 D3=0:
   D5=0:
   D9=0:
   D8=0
2030 IFJ>3THEN2100'BYPASS 3 YR PROP
2040 IFJ=1THEND3=.25*X(10):
   GOTO2100
2050 IFJ=2THEND3=.38*X(10):
   GOTO2100
2060 D3=.37*X(10)
2100 IFJ>5THEN2200'BYPASS 5 YR PROP
2110 IFJ>=3THEND5=.21*X(11):
   GOTO2200
2120 IFJ=1THEND5=.15*X(11):
   GOTO2200
2130 D5=.22*X(11)
2200 IFJ>10THEN2300'BYPASS 10 YR
2210 IFJ>=7THEND9=.09*X(12):
   GOTO2300
2220 IFJ>=4THEND9=0.1*X(12):
   GOTO2300
2230 IFJ=1THEND9=.08*X(12):
   GOTO2300
2240 IFJ=2THEND9=.14*X(12):
   GOTO2300
2250 D9=.12*X(12)
2300 REM ST LINE 15 YR PROPERTY
2310 IF J>15THENRETURN
2320 D8=X(13)/15.
2330 DT=D3+D5+D8+D9
2340 RETURN
9000 REM DATA INPUT
9005 JS=2:
   IFJ1=1THENPG=1
9010 CLS:
   PRINT " ITEM ";TAB(18);"VALUE";TAB(28)"PAG
     E ";PG;"/6"
9020 PRINT STRING$(39,"=")
9040 FOR J=J1TOJ2
9043   IFJ>NTHEN9060
9050   PRINTTAB(2); T$(J);TAB(22);X(J)
9060 NEXT J
9062 IFJ2>NTHENIFJS>2THENJS=2
9070 PRINT @(JS*40)+JR,"->";
9075 PRINT@280,"ARROWMOVE>SHIFTARROWCHANGE PAG
     E  *RET";
9080 Y1$=INKEY$:
   IFY1$=""THEN9080
9090 IFINSTR(A$,Y1$)<>0 THEN 9500
9100 JZ=INSTR(AZ$,Y1$):
   IFJZ=0THEN9080
9110 ON JZ GOTO 9120,9120,9140,9199,9200,9250

9120 PRINT@(JS*40)+JR,"    ";
9122 JS=JS+1:
   IFJS>6THEN JS=2:

9123 IF J2>NTHENIFJS>2THENJS=2
9124 PRINT@(JS*40)+JR,"->";
9130 GOTO 9080
9140 PRINT@(40*JS)+JR,"    ";
9150 JS =JS-1:
   IFJS<2 THEN JS=6
9151 IFJ2>NTHENIFJS>2THENJS=2
9152 PRINT@(40*JS)+JR,"->";:
   GOTO9080
9199 RETURN
```

```
9200 PG=PG+1:
   J1 = J2+1:
   J2=J1+4:
   IFJ1>NTHENJ1=1:
   J2=5
9210 IFPG>7THENPG=1
9220 GOTO 9005
9250 IF J1 = 1 THENPG=1:
   GOTO 9005
9260 J2 = J1-1:
   J1 = J2 - 4:
   PG=PG-1
9270 GOTO 9005
9500 PRINT @(40*JS)+JR,"        ";
9505 PRINT@280,"ENTER REST OF DATA PRESSENTER
     WHEN DONE";
9510 V$=Y1$
9530 PRINT@(40*JS)+JR+4,V$;CHR$(95);"   ";
9540 Y1$=INKEY$:
   IF Y1$ =""THEN 9540
9550 IF Y1$=CHR$(13) THEN 9700
9560 IF Y1$<>CHR$(29) THEN 9600
9570 IF LEN (V$)=0 THEN 9530
9580 V$=LEFT$(V$,LEN(V$)-1):

9590 Y1$="":
   GOTO 9530
9600 IF INSTR(A$,Y1$)=0 THEN 9530
9610 V$ = V$ + Y1$
9620 GOTO 9530
9700 X(J1+JS-2)=VAL(V$)
9710 JS=JS+1
9720 IF JS>6 THEN JS=2
9722 IFJ2>NTHENIFJS>2THENJS=2
9730 GOTO9010
9900 DATA TOTALINVEST,AMT LOAN1$,LIFE YRS,NO.P
     AYMENTS/YR,INTERST RATE,AMT LOAN2$,LIFE
      YRS,NO.PAYMENTS/YR,INTERST#2,3YR PROP,
     5 YR PROPERTY,10YEAR PROPERTY,ST LINE P
     ROP,BASE FOR APPREC,ESTIMATED INCOME,ES
     CAL FOR INCOME,COSTS,ECAL FOR COSTS,APP
     REC RATE
9901 DATA COST OF SELLING,LIFE INVEST,DISCOUNT
      RATE,INFLATION RATE,TAX RATE,CAPGAINS
     TAX,TAXCREDIT1=Y 0=N,END
9910 J=1
9920 READ X$:
   IFX$="END"THENRETURNELSET$(J)=X$
9930 N=J:
   J=J+1:
   GOTO9920
```



Breakeven Analysis

In many business situations the costs can be divided into two parts—fixed costs and variable costs. The *fixed costs,* items such as leases, rents, real estate taxes, etc., are constant regardless of the level of business activity. The *variable costs,* on the other hand, depend on the level of business activity. Generally the variable costs of selling two units of product are twice those involved in selling one unit of product.

The variable costs are costs such as direct labor and parts. The higher the level of business activity, the higher the variable costs. In some situations the variable costs will change as the number of units of product sold increases. In these situations the variable costs are said to follow a *learning curve.* Generally the variable costs decrease as the number of units sold increases.

In many situations it is desirable to know how many units must be sold to break even, i.e., how many units have to be sold to cover all the fixed costs plus all the variable costs of the units sold. The firm neither makes nor loses money at the breakeven point. If the actual number of units sold is less than the breakeven point, the firm has a loss; if the actual number exceeds the breakeven point, the firm has a profit.

This situation for a constant variable cost line is shown graphically in Fig. 5-1. The shaded area represents the area of sales that produces a profit. When the variable costs follow a learning curve, the fixed cost line is a curve. Such a situation is shown in Fig. 5-2. Again the shaded area represents the area where the firm has a profit.

Breakeven analysis is extremely useful because it quantifies the level of sales necessary to cover all costs. If the potential level of sales is less than the breakeven point, there is no hope of making a profit. Thus any investment that promises business activity less than the breakeven point should be avoided. Breakeven analysis can also be used to show where cost cutting can be most beneficial. For example, suppose you can cut fixed costs by 10 percent or variable costs by 1 percent per unit. Which should you do? Breakeven analysis can help you find out.

Quite often it is useful to conduct a breakeven

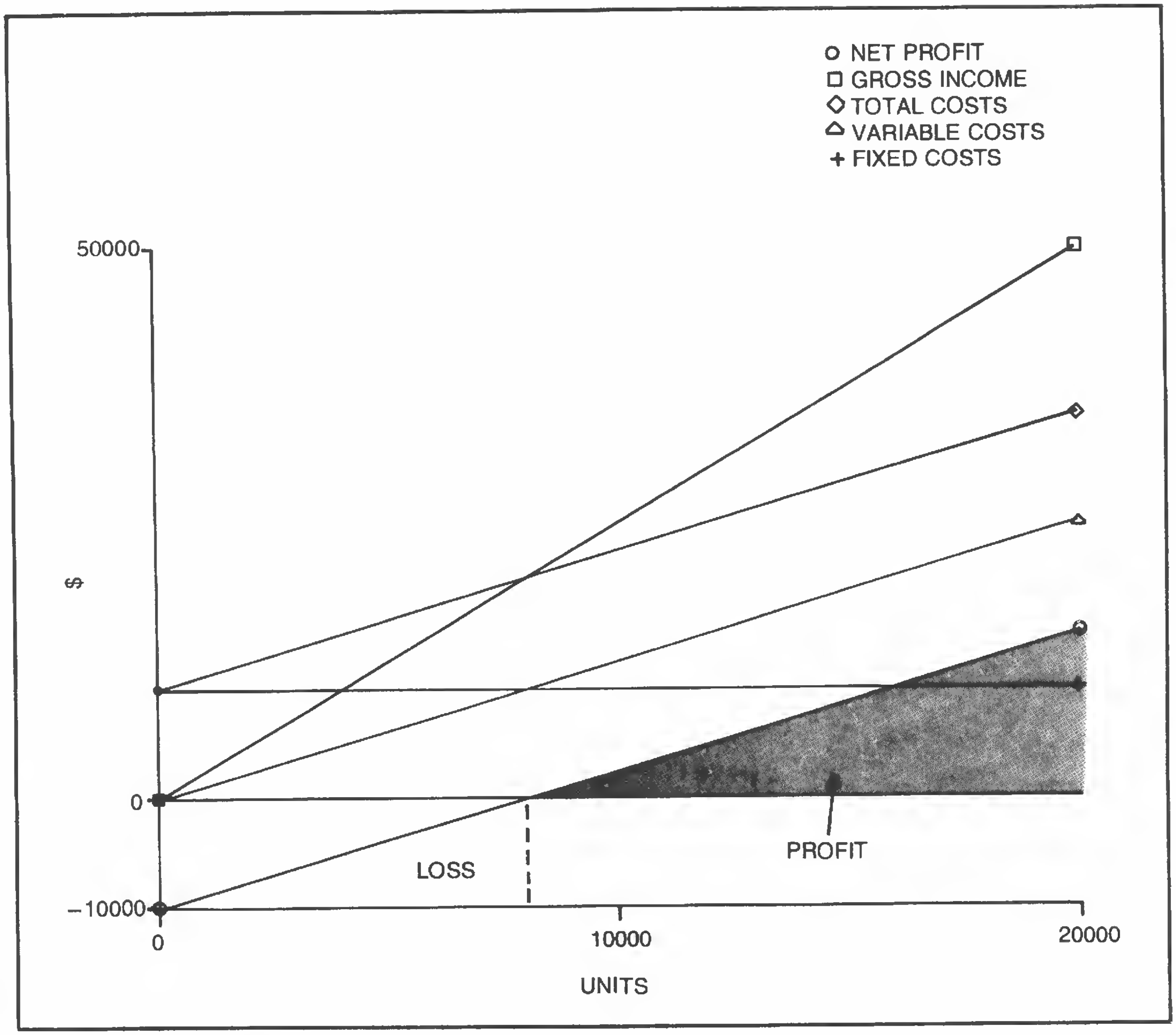


Fig. 5-1. Linear variable cost line.

analysis to determine the impact of the product on cash flow as well as profits. In an analysis based on profits, some costs (such as depreciation), that are not direct out-of-pocket expenses are included in the fixed costs. A breakeven analysis based on cash flow eliminates these costs from consideration and is only concerned with the cash flow generated by the product. *The cash flow breakeven point is always less than the profit breakeven point.*

The program in this chapter, BRKEVN.BA, does all the math involved in breakeven analysis. The program allows you to do the analysis for either a constant variable cost curve or for a learning curve. The learning curve assumes that the costs increase or decrease at a constant rate. You must enter the rate, which may be positive or negative. If the rate is positive, then the variable costs will increase with increased production; if the rate is negative, the variable costs will decrease with increased production.

The program also allows you to specify either a constant selling price or one that changes along with the number of units produced. The rate that the selling price changes by is called the *erosion* *rate*. The erosion rate may be either positive or negative. A negative erosion rate means that the selling price decreases as the number of units sold increases, while a positive erosion rate means that the selling price increases as the number of units sold increases.

The program runs very fast and can provide you with a good picture of the project. You should take full advantage of the speed of the program and use it to run a sensitivity analysis for the project. Try various combinations of costs and selling price to the effects of changes in the business climate and the effects of inaccurate estimates.

The program also plots the results of the analysis on the display. The plot is automatically scaled. The breakeven point is indicated with a tick mark. BRKEVN.BA is listed in Listing 5-1.

For example, suppose you are considering a project that will involve fixed costs of $10,000 per year. The variable costs are initially estimated at $2.00 per unit. Your marketing division estimates that you will be able to sell 10,000 to 40,000 units per year at $3.50 per unit. What is the breakeven point? What is the profit potential for the project?

The run for the example is shown in Fig. 6-3.

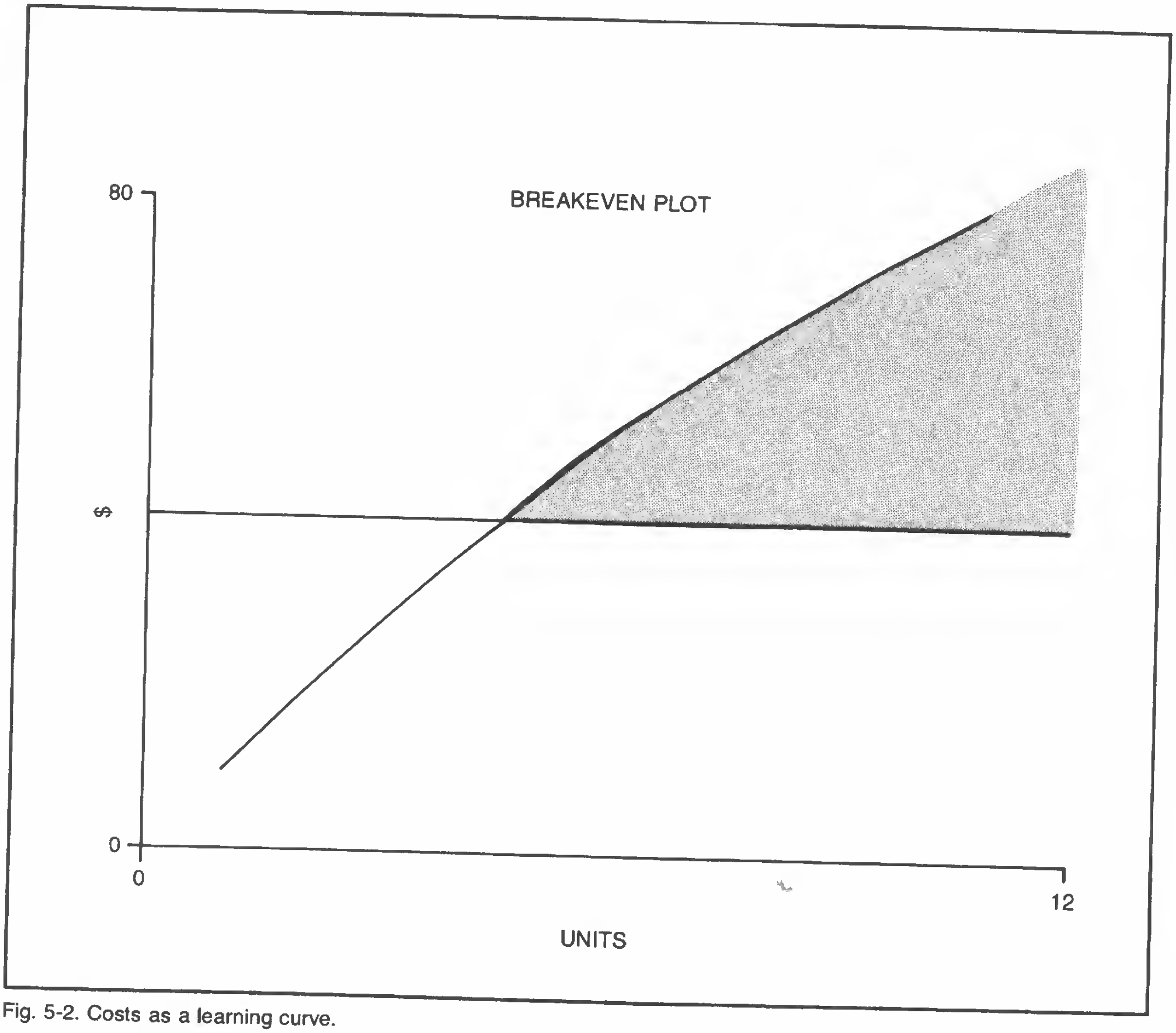
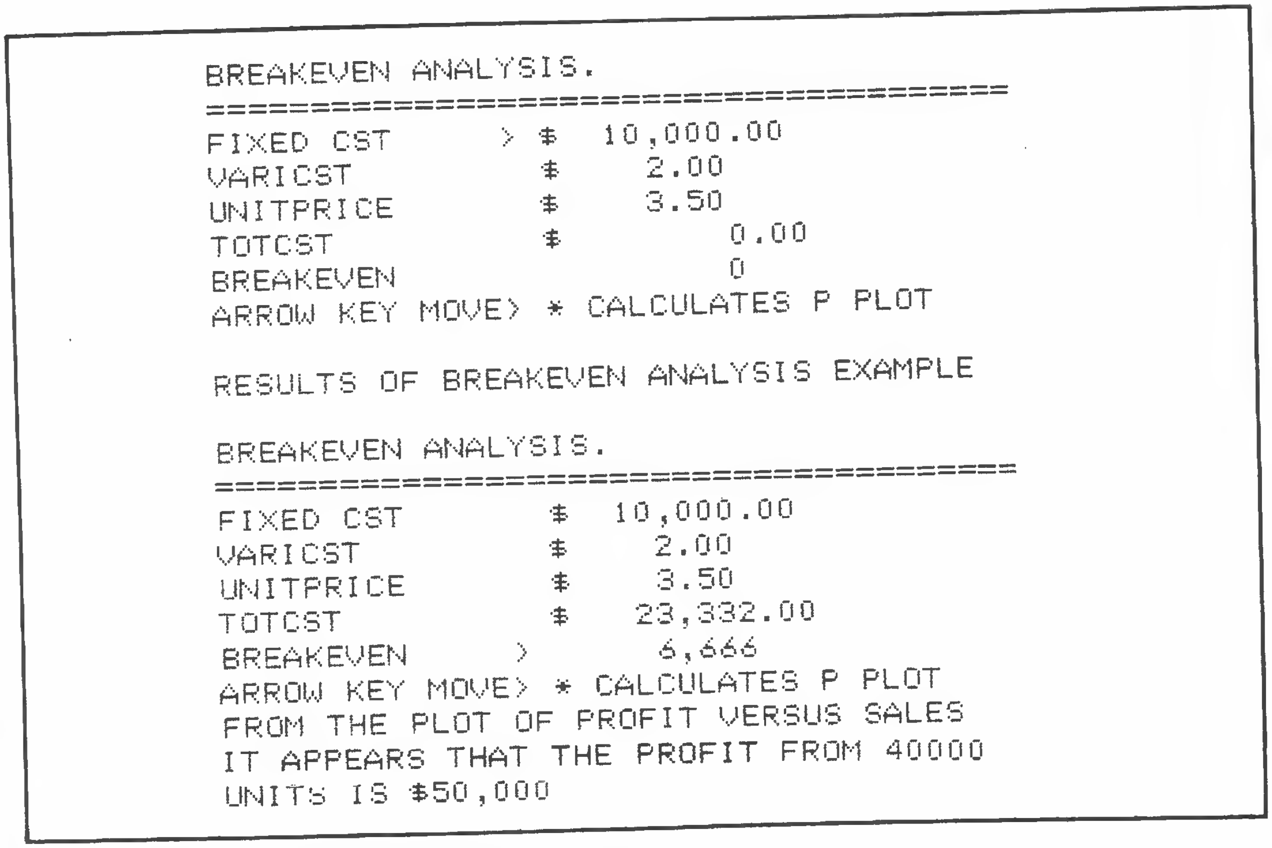


Fig. 5-2. Costs as a learning curve.



```
BREAKEVEN ANALYSIS.
==========================================
FIXED CST        > $  10,000.00
VARICST            $      2.00
UNITPRICE          $      3.50
TOTCST             $            0.00
BREAKEVEN                       0
ARROW KEY MOVE> * CALCULATES P PLOT

RESULTS OF BREAKEVEN ANALYSIS EXAMPLE

BREAKEVEN ANALYSIS.
==========================================
FIXED CST          $  10,000.00
VARICST            $      2.00
UNITPRICE          $      3.50
TOTCST             $    23,332.00
BREAKEVEN        >        6,666
ARROW KEY MOVE> * CALCULATES P PLOT
FROM THE PLOT OF PROFIT VERSUS SALES
IT APPEARS THAT THE PROFIT FROM 40000
UNITS IS $50,000
```

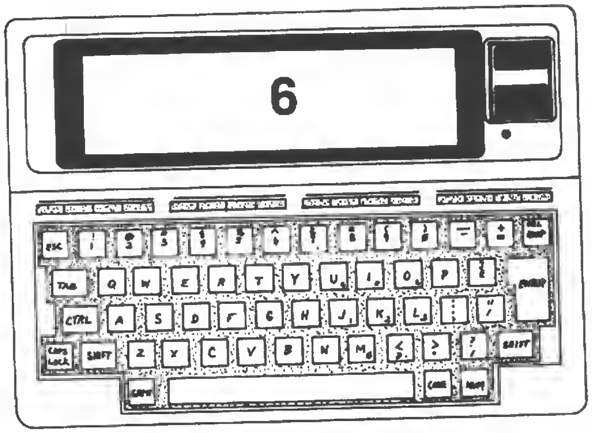
Fig. 5-3. Example run for breakeven analysis using BRKEVN.BA.

Listing 5-1. Breakeven analysis program BRKEVN.BA.

```
10 REM breakeven analysis
20 DIM X(10),Y(10)
30 CLS
51 A$="1234567890.+-"
52 AZ$=CHR$(31)+CHR$(30)+"*"+"Q"
53 J1=0:
   J2 = 4
54 JS=2
55 J9=1
60 GOSUB10000
62 GOSUB 12000
70 CLS:
   GOSUB9000
5000 Z$="":
    CLS:
    XI=X(4)/5:
    XM=10*X(4):
     PRINT"X MAX = ";XM:
     INPUT"ENTER YOUR VALUE OR PRESS ENTER ";Z$
 5002 IFZ$=""THEN5003 ELSE XM=VAL(Z$)
 5003  :
     YI=X(2)*XI-X(1)*XI-X(0):
     YM=X(2)*XM-X(1)*XM-X(0)
 5120 DX=XM-XI:
     DY=YM-YI:
     SX=180/DX:
     SY=54/DY
 5125 X2=.1*DX:
     X0=XI-X2
 5140 REM scale
 5145 CLS
 5150 PRINT@284,XI;:
     L$=STR$(XM):
     L=LEN(L$):
     PRINT@(319-1-L),L$;
 5238 PRINT@293,"UNITS";
 5240 PRINT@240,;:
     PRINTUSING"#####";YI;:
     PRINT@0,;:
     PRINTUSING"#####";YM;
 5245 LINE(34,54)-(214,54):
     LINE (214,54)-(214,56):
     LINE(34,54)-(34,56)
 5246 LINE (34,54)-(34,0)
 5247 X0=X0+X2:
     XP=X0-XI:
     XP=34+XP*SX:
     Y0=X0*X(2)-X(1)*X0-X(0):
     YP=Y0-YI:
     YP=YP*SY:
     YP=54-YP:

 5248 X1=X0+X2:
     XX=X1-XI:
     XX=34+XX*SX:
     Y1=X1*X(2)-X(1)*X1-X(0):
     YY=Y1-YI:
     YY=YY*SY:
     YY=54-YY
 5249 LINE(XP,YP)-(XX,YY)
 5250 IFX1<XMTHEN5247
```

```
5255 REM
5260 X=34:
   LINE(X,54)-(X,56):
   X=X(4)-XI:
   X=33+X*SX:
   LINE(X,54)-(X,50)
5261 Y=0:
   Y=Y-YI:
   Y=SY*Y:
   Y=54-Y:
   LINE (34,Y)-(214,Y)
5270 PRINT@80,"PROFIT$";
5280 Y$=INKEY$:
   IFY$=""THEN5280
5290 RETURN
9000 REM DATA INPUT
9010 CLS:
   PRINT"BREAKEVEN ANALYSIS."
9020 PRINT STRING$(39,"=")
9022 IFJ1<0THENJ1=0:
   J2=4
9040 FOR J=J1TOJ2
9050  PRINTX$(J);TAB(16);:
   PRINTUSINGF$(J);X(J)
9060 NEXT J
9070 PRINT @(JS*40)+14,">";
9075 PRINT@280,"ARROW KEY MOVE> * CALCULATES P
      PLOT";
9080 Y$=INKEY$:
   IFY$=""THEN 9080
9085 IFY$="P"ORY$="p"THENGOSUB5000:
   GOTO9000
9090 IFINSTR(A$,Y$)<>0 THEN 9500
9100 JZ=INSTR(AZ$,Y$):
   IFJZ=0THEN9080
9110 ONJZGOTO9120,9140,9200,9300
9120 PRINT@(JS*40)+14," ";
9122 JS=JS+1:
   IFJS>6THEN JS=2:

9124 PRINT@(JS*40)+14,">";
9130 GOTO 9080
9140 PRINT@(40*JS)+14," ";
9150 JS =JS-1:
   IFJS<2 THEN JS=6
```

```
9152 PRINT@(40*JS)+14,">";:
   GOTO9080
9200 REM CALCULATE
9210 X(4)=INT(X(0)/(X(2)-X(1))):
   X(3)=X(0)+X(4)*X(1)
9230 GOTO 9000
9300 END
9500 PRINT @(40*JS)+14," ";
9505 PRINT@280,"ENTER REST OF DATA PRESSENTER
     WHEN DONE";
9510 V$=Y$
9530 PRINT@(40*JS)+15,V$;CHR$(95);" ";
9540 Y$=INKEY$:
   IF Y$ =""THEN 9540
9550 IF Y$=CHR$(13) THEN 9700
9560 IF Y$<>CHR$(29) THEN 9600
9570 IF LEN (V$)=0 THEN 9530
9580 V$=LEFT$(V$,LEN(V$)-1):

9590 Y$="":
   GOTO 9530
9600 IF INSTR(A$,Y$)=0 THEN 9530
9610 V$ = V$ + Y$
9620 GOTO 9530
9700 X(J1+JS-2)=VAL(V$)
9710 JS=JS+1:
   IFJS>6THENJS=2
9730 GOTO 9010
10000 J=-1
10010 J=J+1:
   READX$:
   IFX$="END"THENJN=J-1:
   RETURN
10020 X$(J)=X$:
   GOTO10010
10030 DATA FIXED CST,VARICST,UNITPRICE,TOTCST,
     BREAKEVEN,END
12000 FORJ=0TO4:
   READF$(J):
   NEXTJ:
   RETURN
12010 DATA"$##,#####.##","$##,##.##","$#,###.#
    #","$##,######.##","#,########"
```



Ratio Analysis

The financial statements of a company, namely the balance sheet and the income statement, are the prime source of information for financial decision making. Sound financial decisions depend on the correct interpretation of this information. Thus, financial statement analysis is an important facet of the process.

The reasons for financial statement analysis are many. For example, you may be considering making a loan either as a short term creditor or as a long term bond holder. Or you might be considering the purchase of stock in a company. Or you might be the manager concerned with the future of the company. Or finally, you might be the sole owner or partner of a business. In all of these situations you must make decisions based on the information contained in the financial statements.

The factors about a company that you judge to be important will depend on the situation. A short term creditor, for example, will be interested in the company's ability to pay over the short term. An owner, on the other hand, will be interested in the efficient utilization of the company's funds. It is possible that conditions that an owner finds favorable will be considered unfavorable by a creditor. Over the past several years, creditors, owners and managers have discovered that the information in the financial statements was often hard to interpret. Thus they developed the idea of calculating various ratios to show how various items in the financial statements were related to one another and to the general health of the company. The ratios also made it easier to compare the health of various firms with one another and with the spectrum of similar firms.

Many categories of ratios have been found useful over time. *Liquidity ratios* are used to determine the ability of a company to meet its near-term debts. *Activity ratios,* also called *asset management ratios,* are used to determine the efficiency of asset usage. *Leverage ratios,* sometimes called *safety ratios,* are used to determine the use of debt financing. *Profitability ratios* are used to determine the degree to which business activities result in profit. And, finally, *market ratios* are used to deter-

Table 6-1. Some Common Ratios.

Type	Measures	Common examples
Liquidity	Ability to pay bills.	Current ratio Acid test ratio
Asset Management	Efficiency of asset use.	Average Collection Period Inventory turnover
Leverage	Risk of business	Debt to asset ratio Times interest earned Fixed charge coverage
Profitability	Profitability	Profit margin Return on assets Return on equity

mine how investors see the company. Some of the most common ratios and their uses are shown in Table 6-1.

Although the ratios themselves are important, the trends of the various ratios over time are often even more important. For example, a company with a trend of improving profitability ratios is more interesting than another company with a decreasing trend of profitability ratios.

The program presented in this chapter allows you to enter data from the balance sheet and income statement for several time periods (or a single time period if you desire). The program then calculates and displays the various ratios for each time period. An option to plot the ratios is also provided.

LIQUIDITY RATIOS

As stated above, liquidity ratios are used to evaluate the ability to meet short term debt payments. The current ratio and the acid test ratio are the two major liquidity ratios.

The *current ratio* (used to measure the ability to cover near term debt) is defined as current assets divided by current liabilities. The most common current assets are cash, marketable securities, prepaid expenses, accounts receivable, and inventory. The usual current liabilities are accounts payable, short-term loans (or notes payable), the current portion (due within the next 12 months) of the long term debt, and accrued expenses (usually wages and taxes). The basic assumption of the current ratio is that if things go bad, the assets can be sold and used to cover the liabilities. Of course this assumes that the assets are really liquid and can be sold for their book value.

A current ratio of 1.0 says that the firm can just meet its bills while a ratio of 2.0 says that the firm has a cushion to meet any unforeseen needs. Generally a current ratio of 2.0 shows that the firm is in good shape. However, the exact value of the ratio that should be considered good or bad depends in great part on the type of business. For example, a fast-food restaurant might have a current ratio of less than 1 and still be in fair shape. Service industries also tend to have weak current ratios, so be sure to look at the type of business in analyzing this measurement.

Quite often a bank or other creditor will make a loan with the stipulation that the current ratio not drop below a given value. If the current ratio does drop below the limit, the loan is considered to be in default and payable at once.

The question of liquidity of assets, especially inventory, gave rise to the acid test or quick ratio. The *acid test ratio* is current assets minus inventory, divided by current liabilities. The acid test ratio is a more conservative measure of the ability to meet near-term obligations than is the current ratio. The rationale for neglecting inventory is that inventory is the hardest of the assets to convert to cash. In fact quite often inventory cannot be liquidated for the value shown on the books—especially if cash is needed quickly. The *quick ratio* basically answers the question, "If all sales disappear, could the firm meet its current obligations with the liquid assets on hand?"

A reasonable value of the acid test ratio again depends on the type of business. For example, a manufacturer should have an acid test ratio near unity. A retail business, on the other hand, will usually have an acid test ratio of less than 1.0. A creditor would like to see the acid test ratio have a value greater than 1.0.

Creditors, owners, and managers look at the liquidity ratios through different eyes. The creditor wants to be sure that the ratios show financial strength and the ability to pay bills. The owner, on the other hand, wants to be sure that the liquidity ratios are not too high. High liquidity ratios indicate either that the assets are not really liquid or that the assets are not profitability used.

In short, if the liquidity ratios are too high or too low, corrective action is needed. The exact steps to be taken depend on the situation and require careful study.

A creditor would like to see either a constant or an increasing trend in the current and acid test ratios. A decreasing trend in the liquidity ratios could be a sign of future problems. Such a trend would be reason to be wary of advancing too much credit to the company.

LEVERAGE RATIOS

A company has two sources of funds—debt and equity. Debt requires that the company pay a fixed cost for funds. If the business activity financed by the debt returns more than the cost of borrowing the money, the owners get the excess. Of course, if the business financed by the debt returns less than the cost of borrowing the money, the owners are stuck with the shortfall. Debt increases the risk of doing business.

The leverage ratios indicate the extent to which a company uses debt and the ability of the company to meet debt obligations. The three leverage ratios that we'll discuss are the *debt ratio*, the *times interest earned*, and *fixed charge coverage*.

Debt Ratio. The debt ratio is one of the most common of the leverage ratios. This ratio shows how much of the firm's assets are financed by debt. The ratio tells a lot about the ability of the firm to meet both its long- and short-term debt obligations and is a good indication of the risk involved in lending to the firm.

The debt ratio is calculated by dividing the total liabilities by total assets. Total liabilities means all debt, i.e., all current liabilities as well as long-term debt such as bonds and mortgages. The information for calculating the debt ratio can be found on the balance sheet.

The debt ratio is a measure of the riskiness of the firm. A high ratio indicates that the owners' may be overextended and that the firm would have a difficult time surviving a period of adverse business. Lenders and suppliers desire a low value of the debt ratio—the lower the better. A firm will often find that if the debt ratio gets too high lenders and suppliers will curtail credit. Lenders might call notes and suppliers might require cash before goods are delivered.

The exact value of the debt ratio that indicates a problem depends on several factors. For example, a debt ratio of 2.0 might indicate a serious problem for a new business, yet the same ratio would not be of concern to a well-established firm. Some industries, such as the utility industry, have debt ratios greater than 3.0 yet are not considered risky.

An owner likes to see a fairly high value of the debt ratio. High values indicate that the owners' funds are highly leveraged with other people's money. A manager tends to desire a debt ratio somewhere between the low ratio desired by bankers and the high ratio desired by owners.

The debt-to-worth or debt-to-equity ratio is another often used ratio. This ratio is similar to the debt ratio except that the liabilities are divided by the total owners' equity instead of the total assets. The debt-to-worth ratio is interpreted the same as the debt ratio.

Times Interest Earned. *Times interest earned* measures how much of the firm's earnings are required to pay the interest on outstanding loans. This ratio is calculated by dividing the operating income by the interest expense. The operating income is calculated by subtracting the cost of goods sold from the sales income. Note that interest expense is not deducted from the revenue.

A low value of the times interest earned is a danger signal. If the times interest earned is less than 1.0, for example, the firm is not earning enough to even pay the interest due on its debt. There is no money left over to provide for profit to the owners, bonuses to the managers and employees, or principal payments on the loans. Thus everyone—creditors, owners, and managers—wants to see a high value of times interest earned.

Fixed Charge Coverage. A variation of the times interest earned ratio is the *fixed charge coverage* ratio. This ratio measures the ability of the firm to meet its fixed charge obligations. A high value indicates that the firm is in good shape and that it can probably weather a period of poor business. A low value of the fixed charge coverage ratio, on the other hand, is a sign of trouble and risk.

The fixed charge coverage is calculated by dividing the fixed charges for the period by the earnings for the period. If the ratio is less than unity, the firm's earnings are not enough to cover all the fixed charges. A firm might be able to afford short-term periods when the fixed charge ratio is less than unity. But no firm can survive several periods of unacceptable fixed charge ratios.

Creditors, managers, and owners look at the fixed charge ratio in the same way as they look at the times interest earned ratio. They all want to see the fixed charge ratio have a high value.

ACTIVITY RATIOS

The activity ratios tell how efficiently the firm is using its assets. The most common activity ratios are the *inventory turnover*, the *average collection period*, and the *total asset turnover*. These three ratios are good measures of the efficiency of a firm's management.

Inventory Turnover Ratio. The inventory turnover ratio is used to measure the efficiency of inventory usage. The inventory turnover ratio is a more powerful tool for measuring the efficiency of inventory management than the absolute value of the inventory. One of the reasons it is powerful is that it combines information from both the balance sheet and the income statement. It is calculated by dividing the cost of goods sold (from the income statement) by the average inventory (from the balance sheet).

Because the inventory can account for more than half of the firm's current assets, the inventory turnover ratio is especially important. Another reason for keeping close watch on this ratio is that inventory is a risky asset. Sudden changes in the business climate or technology can reduce the value of inventory to zero. (Ask the people who had large inventories of slide rules a few years ago.)

Generally, a high value of inventory turnover is desired. Low values of inventory turnover may indicate that inventory is unsalable or greatly overvalued. A low value of the inventory turnover ratio also indicates that much of the firm's assets are tied up in inventory. The trend of inventory turnover ratio over time is also important. A trend that shows a steady decrease in the inventory turnover ratio, for example, is a likely indication of future trouble. Such a trend indicates that goods are not selling as fast as they once did. Action should be taken to find out why inventory is building up. Perhaps the demand for the goods has dropped. (We might be selling hula hoops.)

A very high value of the inventory turnover ratio may indicate problems of another kind. Our business may have excessive stock out and backorders. Perhaps the inventory is not large enough to support the level of business activity. Again the trends of the turnover with time are important. A steady increase in the inventory turnover ratio may be good, or it may indicate that inventory is too small.

Average Collection Period. The average collection period measures the effectiveness of a firm's credit management. It is almost impossible to do business without extending credit to customers. Therefore it is essential to collect outstanding bills efficiently. Failure to do so can result in poor cash flow and other problems with the business. The average collection period is calculated from the following formula

$$\text{average collection period} = 365 \times \left(\frac{\text{accounts receivable}}{\text{sales}} \right)$$

The value of the average collection period depends on industry practice and your credit policy. Industry practice generally determines credit terms—discounts for example, and how long before bills must be paid. These general credit terms are usually outside your control. If you do not conform to industry practice, you may not be able to compete effectively. You do have control over your collection policy. Collection policy determines how vigorously you try to collect outstanding bills. You also determine who gets credit.

The average collection period then measures the quality of your credit customers and the efficiency of your credit department. If the average collection period is unacceptable, you need to work on both your credit policy and your collection efforts.

In general, the average collection period should not exceed 1.3 times the credit period you give your customers. For example, if you extend 30-day terms to your customers, your average collection period should be 1.3×30 or 39 days. If the collection period is too high, you may be extending credit to customers who do not deserve credit. A high collection period also indicates that your collection efforts are too lax. Look at your billing and collection practices to make sure that you bill customers at once and that you vigorously pursue overdue accounts.

On the other hand, if your collection period is too low you may be passing up good credit sales and reducing the potential profit of the business. As you might expect everyone—creditors, owners, and managers—want to see a low average collection period.

Total Asset Turnover. The total asset turnover is an overall indicator of the efficiency of asset usage. The higher the ratio, the more efficiently assets are being used to produce sales—and thus profits.

The total asset turnover is calculated by dividing sales by total assets. It tells you how rapidly assets are converted to sales. High values of the total asset turnover indicate efficient utilization of assets to produce sales. Low values may indicate that the assets are overvalued and perhaps outmoded. A firm with poor utilization of its assets can not remain profitable for long.

PROFITABILITY RATIOS

Activity ratios are measures of the cash flow of a business. While cash flow is important, profit is what really matters—at least in the long run. The next set of ratios we will look at measure the profitability of a firm. These ratios are of special interest to investors because they predict probable return on investment. The ratios we will look at are the *net profit margin,* the *net return on assets,* the *operating profit margin,* the *operating return on assets,* and the *return on net worth.*

Net Profit Margin. The net profit margin measures the ability of a firm to convert sales into after-tax profits. The formula is

$$\text{net profit margin} = \frac{\text{profit after tax}}{\text{sales}}$$

Note that the profit should not include extraordinary items. (Extraordinary items are revenues or expenses that do not normally occur in the normal course of business. Examples are selling land, settling major lawsuits, etc.) The profit margin measures the amount left over for the owners after all expenses are paid. Thus owners want to see it as high as possible.

Acceptable values of the profit margin depend a great deal on the type of business. Some businesses have high volume and low profit margins. Others have just the opposite. In general, the profit margin of the firm you are interested in should be in line with that of other, similar firms.

If the ratio is lower than expected, corrective action must be taken. Likely places to look for the source of the problem are expenses, pricing policy, and sales volume. The other ratios can often help you pinpoint the reason for the problem. The ratios can also help you find the best type of corrective action.

The trends of net profit margin should be consistent over time. Net profit margins that change drastically from time period to time period indicate a poorly managed company. In fact, inconsistent profit margins are more likely to indicate problems than do consistently low profit margins.

As you might expect, everyone—owners, managers and creditors—wants to see good profit margins. Good profit margins indicate that the firm is likely to be in business for a long time, and thus credit can be extended without fear of loss.

Net Return on Assets. The net return on assets measures how productive your assets are. It measures how much profit is generated by the assets. This ratio is used for making financial decisions ranging from pricing to investing.

The net return on assets is often looked at as "the bottom line." Management that produces high values for the ratio is looked upon with favor, and management producing low values is fired.

The net return on assets is calculated from the following formula:

$$\text{net return on assets} = \frac{\text{after tax profit}}{\text{total assets}}$$

If net return on assets is too low, it may indicate that profits are too low, that too much has been invested in assets, or that the assets are overvalued. Corrective action may involve boosting sales and profits, or it may involve selling assets. Well-managed companies often do both.

Very high values of net return on assets may also indicate problems. For example, the current level of assets may not support future growth of the business. High values of the net return on assets may also indicate that older assets need to be replaced in the near future.

The time trend of net return on assets should be stable or increasing. Because return on assets depends both on the level of profits and on the level of assets, care must be taken to determine the reasons for an increasing trend. An increasing trend due to improved profits is very different than an increasing trend due to reduced assets. The former indicates a business with a long term future, while the latter may indicate a business with a very short future—or at least a business that will require major investments in assets soon. A stable net return on assets combined with a good debt ratio indicates the firm can repay its bills and is thus a good credit risk.

Operating Profit Margin. The operating profit margin measures the ability to use sales to generate profits. It tells how effective a company's sales are for producing profits. The operating profit margin is calculated using the formula below:

$$\text{operating profit margin} = \frac{\text{net pretax operating income}}{\text{sales}}$$

As was the case of net profit margin, the income should not include extraordinary items. The purpose of the ratio is to measure the efficiency of the normal business activity; inclusion of extraordinary items distorts the measurement.

The interpretation of the operating profit margin is the same as the interpretation of the net profit margin. Everyone wants to see a high value for the ratio.

Return on Net Worth. For the owners of a business the bottom line is return on net worth or return on equity. This ratio measures how much you, as an owner, earned on your investment. This ratio provides the bottom-line measure of everything—management efficiency, pricing, credit policy, cost control, asset management, and so on.

This is the ratio of most importance to the owners. If it is unsatisfactory, one of the other ratios may help determine the reasons for poor performance. The ratio should be high enough to compensate the owners for the risk of the business. If it is too low, the owners would be better off getting out of the business and putting their money into another venture.

Creditors are not as interested in the return on net worth as the owners are. Creditors are interested in the ability of the company to pay its bills—not on how good an investment it is for the owners. However, very low values of return on net worth are often taken as a sign of poor management and may reduce the credit extended to a firm.

PRICE-EARNINGS RATIO

The last ratio we will look at is the price-earnings (P/E) ratio. This is one of a set of ratios called *market ratios.* The price earning ratio measures the reaction of investors to the company's performance. High price-earnings ratios indicate that investors expect the company to grow. Low price-earnings ratios indicate that investors are not too optimistic about the company's prospects.

Firms with high P/E ratios often find it easy to raise money in the capital markets, while those with low P/Es find it difficult to raise money.

Many books on the stock market suggest that high price-earnings ratios are a sign of an overvalued company and thus represent risky investments.

THE PROGRAM

Now that we have defined the ratios, let's look at the program that does all the calculations for us. RATIO.BA (Listing 6-1) calculates ratios shown in Table 6-2, using the data in Table 6-3.

Except for those items pertaining to the common stock, all of the data can be found either on the balance sheet or the income statement. The data can be entered for one or for several time periods.

The program consists of three modules—a data entry module, a calculation module, and a plotting module. (The plotting module is actually a separate program.) The data entry module is based on the master data entry module. Because there are more data than can be shown on the display at one time, a provision for moving to new data pages must be provided. This is handled by using the shift arrow keys to move from page to page. The shift left and right arrows are used to move from time period to time period, each screen page displaying three time periods. The shift right arrow key moves forward while the shift left arrow key moves backwards. For example, if the screen displays data for 1972, 1973, and 1974, the shift right arrow will move to display 1975, 1976, and 1977. The shift left arrow will show 1969, 1970, 1971 (if they exist). The shift up and down arrows move from data item to data item.

Table 6-2. Ratios Calculated by RATIO.BA.

Current Ratio

Acid Test Ratio

Debt Ratio

Times Interest Earned

Fixed Charge Coverage

Inventory Turnover

Average Collection period

Total Asset Turnover

Net Profit Margin

Net Return on Asset

Return on Net Worth

Price Earnings Ratio

Table 6-3. Input Data Required by RATIO.BA.

Sales

Cost of Goods Sold

Gross Profit (calculated by program)

Depreciation

Overhead

Sales Expense

Net Operating Income (calculated by program)

Interest

Earnings (calculated by program)

Tax

Net Income (calculated by program)

Earnings/Share

Cash

Accounts Receivable

Inventory

Fixed Assets

Total Assets (calculated by program)

Accounts Payable

Miscellaneous Accruals

Total Current Liabilities (calculated by program)

Long-Term Debt

Common Stock

Paid-in Capital

Retained Earnings

Total Liabilities and Net Worth (calculated by program)

Lease Payments

Debt Payment

Price of Common Stock

The unshifted left and right arrow keys move from time period to time period; the unshifted up and down arrows move from data item to data item *within* a single time period. When you are entering data, the cursor automatically moves to the next time period. Also note that the cursor "wraps around," i.e., it moves from the last time period or data item back to the first. This is consistent with the way the data entry module works for all the programs in this book.

When you have entered and checked all the data, you can begin calculations by pressing the * key—the convention used throughout this book. The calculated answers are displayed in the same format as was used for data entry. You can change pages using the shift arrow keys.

If you want to plot the ratios, press P. You will be asked for a filename under which to save the results. The results are then written to the file and the plot program loaded. You will then be asked for the file you want to plot. The program will load the file and plot each of the ratios, with each plot automatically scaled. Press any key to move to the next plot.

The plot program is a separate program in order to conserve memory. With the main ratio program, memory must be available for all the input data and the various ratios. The plot program does not need all the input data. Thus considerable memory can be saved by storing the ratios in a RAM file, discarding the input data, and then plotting the data from the RAM file. I found that 32K would be required to plot ratios if the plotting program were written as a subroutine in the main ratio program, but 24K is plenty of space for it as a separate program. The plotting routine is a useful program to have; it is discussed in more detail in Appendix A.

Be sure to save the program to tape as soon as you have it typed into your computer.

EXAMPLE OF RATIO.BA

You are the president of a company supplying materials to the widget industry. You are considering the advisability of extending credit to the XY Widget Company. You have requested that the XY Widget Company supply you with their balance sheets and income statements for the past four years. You also obtained data on their common stock. The balance sheet for the four years is given in Table 6-4, the income statement is given in Table 6-5, and some additional data may be found in Table 6-6. Use the RATIO.BA program to analyze these data.

Turn on the Model 100, enter BASIC, and type RUN"RATIO.BA. Enter the number of periods by typing 4 and pressing ENTER. The periods for which we have data are 1979, 1980, 1981, and 1982, so we name the four periods 1979, 1980, 1981, and 1982. The display is then as given in Fig. 6-1. We are now ready to fill in the data. The first page of the data entry form is given in Fig. 6-2.

We will fill the form in by first entering all the data for 1979, 1980, and 1981 (the first horizontal

Table 6-4. Example Balance Sheet Data for RATIO.BA.

Balance Sheet for XY Widget Company

Item	1979	1980	1981	1982
Cash	100,000	110,000	130,000	120,000
Accounts receivable	80,000	90,000	90,000	100,000
Inventory	50,000	65,000	60,000	70,000
Current Assets	230,000	265,000	280,000	290,000
Fixed Assets	345,000	345,000	400,000	450,000
Total Assets	575,000	610,000	680,000	740,000
Accounts Payable	120,000	145,000	150,000	200,000
Miscellaneous Accruals				
Total Current Liabilities	120,000	145,000	150,000	200,000
Long-term Liabilities	200,000	200,000	300,000	360,000
Common Stock	500,000	500,000	500,000	500,000
Paid-in Capital	400,000	400,000	450,000	450,000
Retained Earnings	9,000	10,000	9,000	8,000
Total Liabilities & Net Worth	1,229,000	1,305,000	1,409,000	1,518,000

Table 6-5. Example Income Statement Data for RATIO.BA.

Income Statement XY Widget Company

Item	1979	1980	1981	1982
Sales	400,000	420,000	450,000	430,000
Cost of Goods Sold	115,000	120,000	130,000	130,000
Gross Profit	285,000	300,000	320,000	300,000
Depreciation	20,000	20,000	20,000	20,000
Overhead	190,000	195,000	200,000	205,000
Sales Expense	25,000	30,000	40,000	30,000
Net Operating Income	50,000	55,000	60,000	45,000
Interest	20,000	20,000	20,000	25,000
Earnings	30,000	35,000	40,000	20,000
Tax	15,000	17,500	20,000	10,000
Net Income	15,000	17,500	20,000	10,000
Earnings/share	1	1.17	1.33	0.67

Table 6-6. Other Information for RATIO.BA Example.

Other Information for XY Widget Company

Item	1979	1980	1981	1982
Lease Payments	100,000	100,000	120,000	120,000
Debt Payments	80,000	80,000	90,000	90,110
Price	10	12	12	12

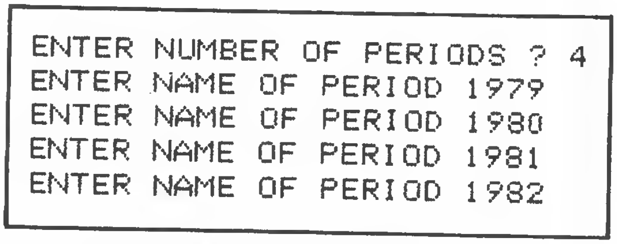
page). After all the data for these years have been entered, we will enter the data for 1982. The form should begin to look like Fig. 6-3. Note that the cursor automatically moves from column to column as data are entered. When you have entered all the data for the first page, press shift down arrow and get the form shown in Fig. 6-4. Fill this form in as shown in Fig. 6-5 and go to the next page by pressing the down arrow to get Fig. 6-6. Fill in this one too, and continue through the data entry (Fig. 6-7).

Now we are ready to enter the data for 1982. First press shift down arrow to move back to the first page. (We could have used shift up arrow twice to do the same thing. However, it is faster to take advantage of the wrap-around feature built into the program to move directly from the last page to the first page.)

When you're back to the first page, press shift right arrow to move to Fig. 6-8. Now fill in the data for 1982. When page 1 is filled in, press shift down arrow to move to page 2 and so on. When all the data are entered, press * to start the calculations. The results are shown in Fig. 6-9.

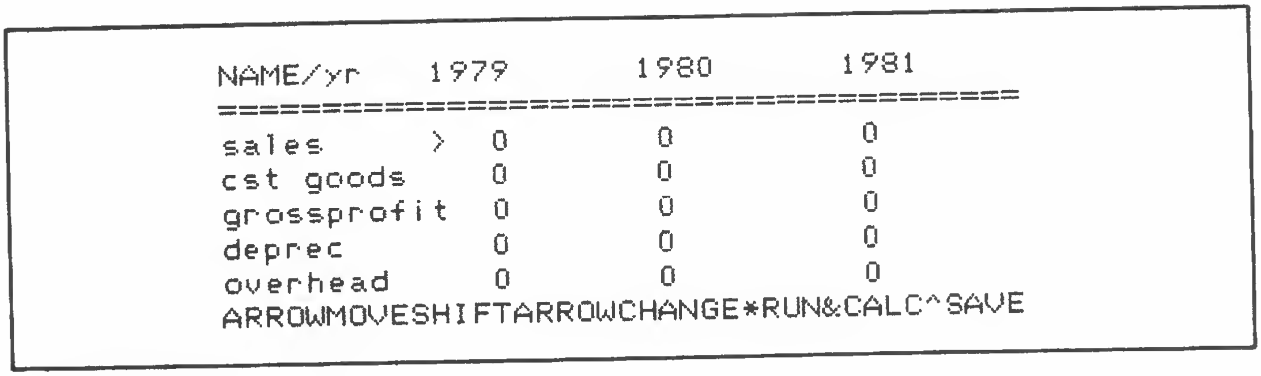
After all the calculations are completed, press P to plot the data. When the program asks for a file name use TEST. Then tell the program that you want to store the data in a RAM file. The data will be stored and the program will end.

Now run PLOT and tell the computer that the file name is TEST and that the file is in RAM. The



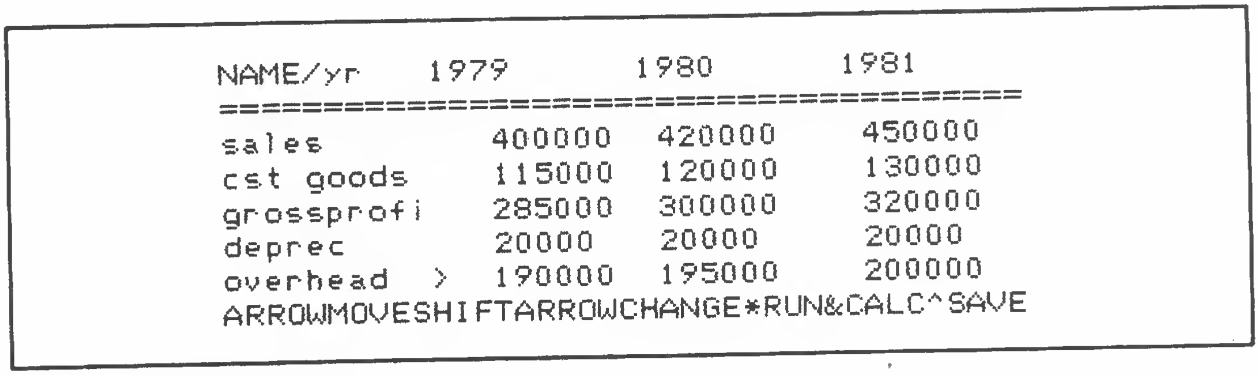
```
ENTER NUMBER OF PERIODS ? 4
ENTER NAME OF PERIOD 1979
ENTER NAME OF PERIOD 1980
ENTER NAME OF PERIOD 1981
ENTER NAME OF PERIOD 1982
```

Fig. 6-1. Initial screen for ratio analysis.



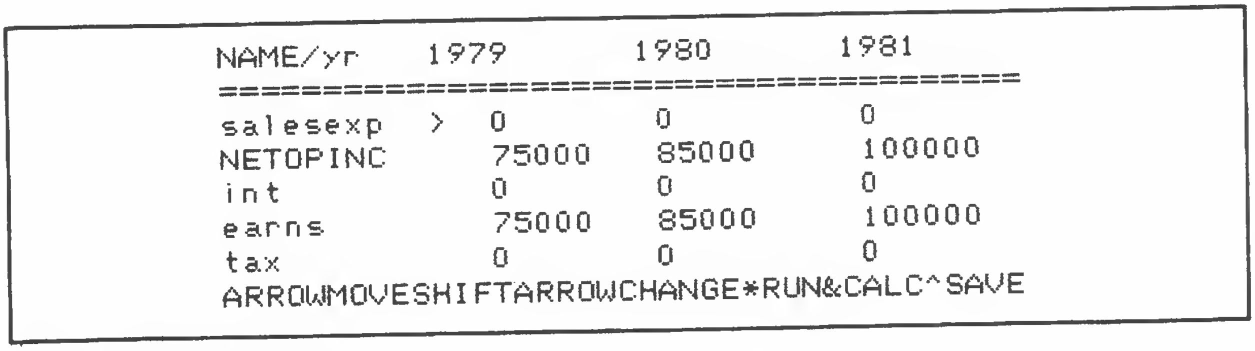
```
NAME/yr    1979        1980        1981
=========================================
sales      >  0         0           0
cst goods     0         0           0
grossprofit   0         0           0
deprec        0         0           0
overhead      0         0           0
ARROWMOVESHIFTARROWCHANGE*RUN&CALC^SAVE
```

Fig. 6-2. First page of data entry.



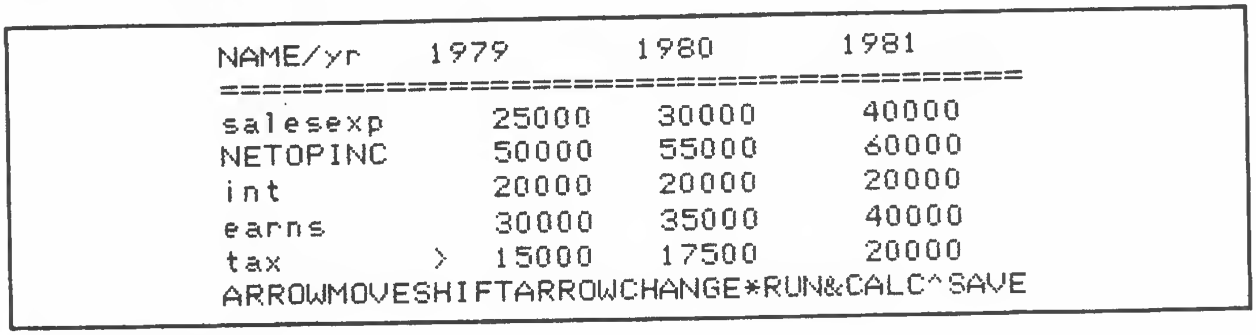
```
NAME/yr    1979        1980        1981
=========================================
sales           400000   420000     450000
cst goods       115000   120000     130000
grossprofi      285000   300000     320000
deprec          20000    20000      20000
overhead   >    190000   195000     200000
ARROWMOVESHIFTARROWCHANGE*RUN&CALC^SAVE
```

Fig. 6-3. Continuing the data entry.



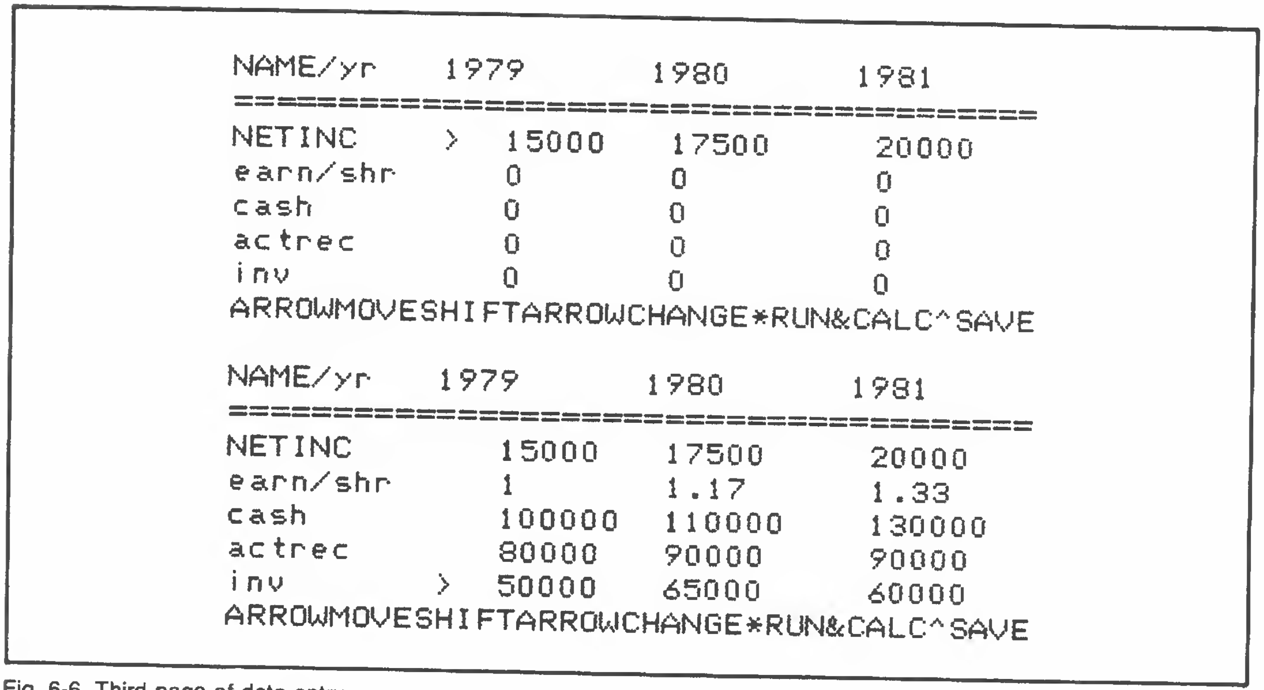
```
NAME/yr    1979        1980        1981
=========================================
salesexp   >  0         0           0
NETOPINC      75000     85000       100000
int           0         0           0
earns         75000     85000       100000
tax           0         0           0
ARROWMOVESHIFTARROWCHANGE*RUN&CALC^SAVE
```

Fig. 6-4. Second page of data entry.



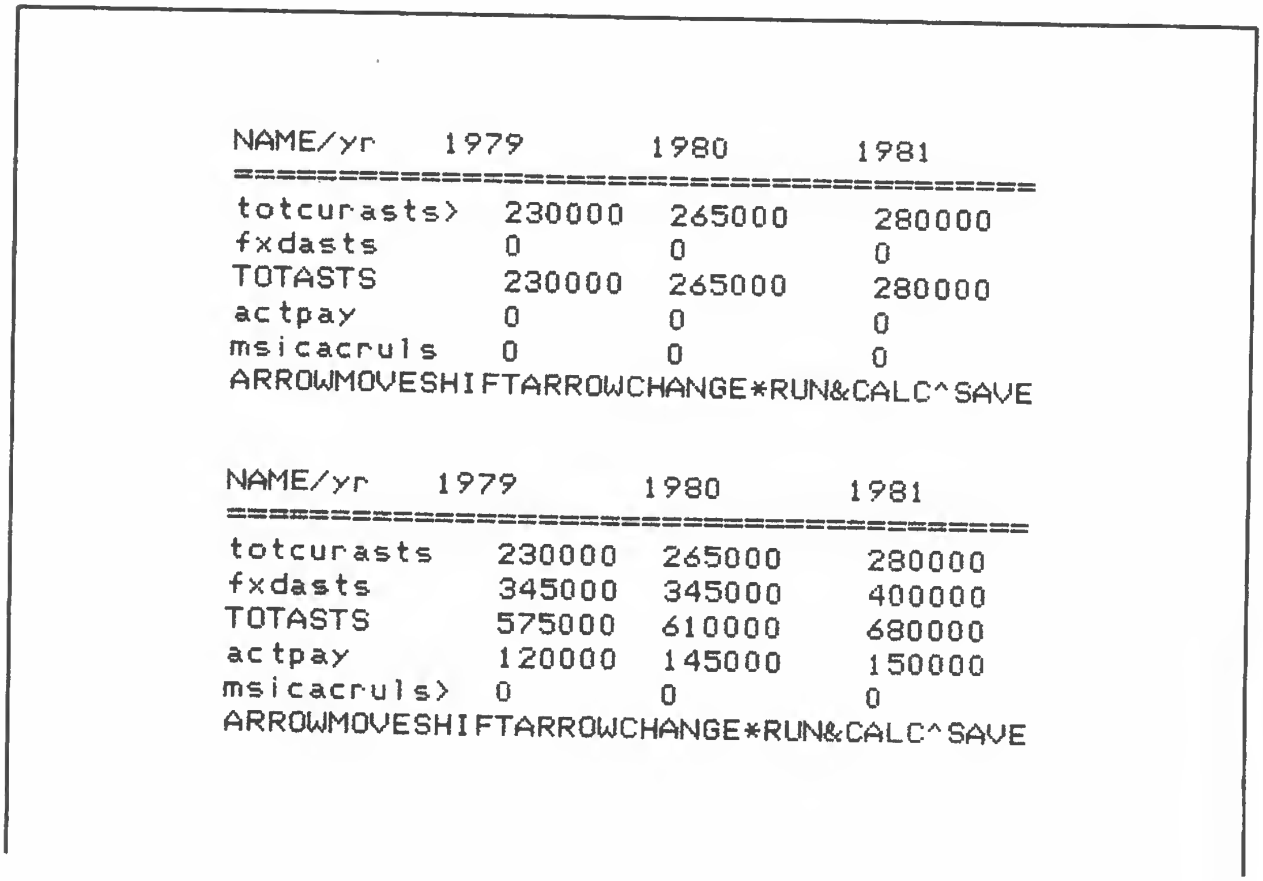
```
NAME/yr    1979        1980        1981
=========================================
salesexp      25000     30000       40000
NETOPINC      50000     55000       60000
int           20000     20000       20000
earns         30000     35000       40000
tax        >  15000     17500       20000
ARROWMOVESHIFTARROWCHANGE*RUN&CALC^SAVE
```

Fig. 6-5. Filling in the form.

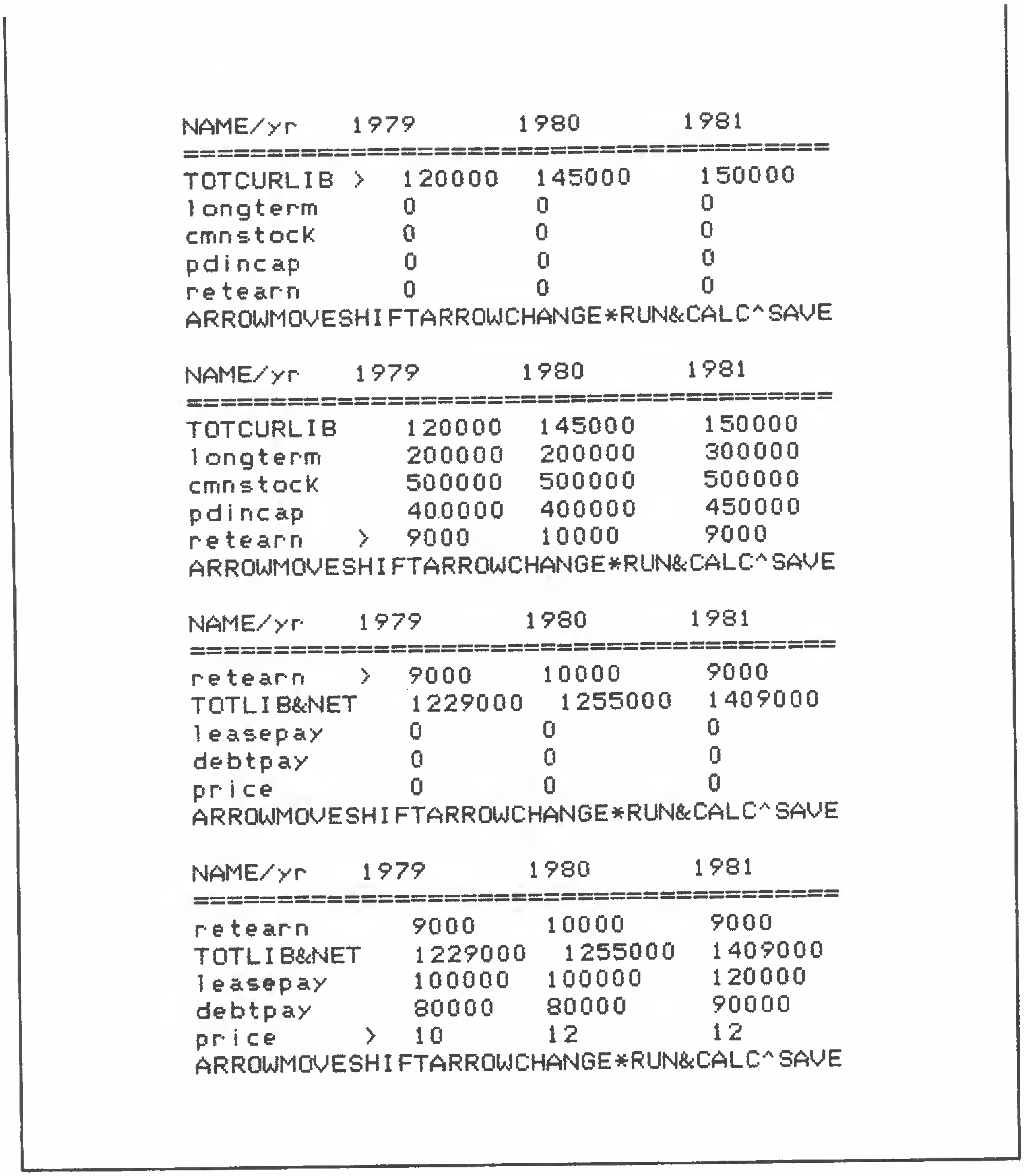


```
NAME/yr    1979        1980        1981
=======================================
NETINC     > 15000     17500       20000
earn/shr     0         0           0
cash         0         0           0
actrec       0         0           0
inv          0         0           0
ARROWMOVESHIFTARROWCHANGE*RUN&CALC^SAVE

NAME/yr    1979        1980        1981
=======================================
NETINC       15000     17500       20000
earn/shr     1         1.17        1.33
cash         100000    110000      130000
actrec       80000     90000       90000
inv        > 50000     65000       60000
ARROWMOVESHIFTARROWCHANGE*RUN&CALC^SAVE
```

Fig. 6-6. Third page of data entry.



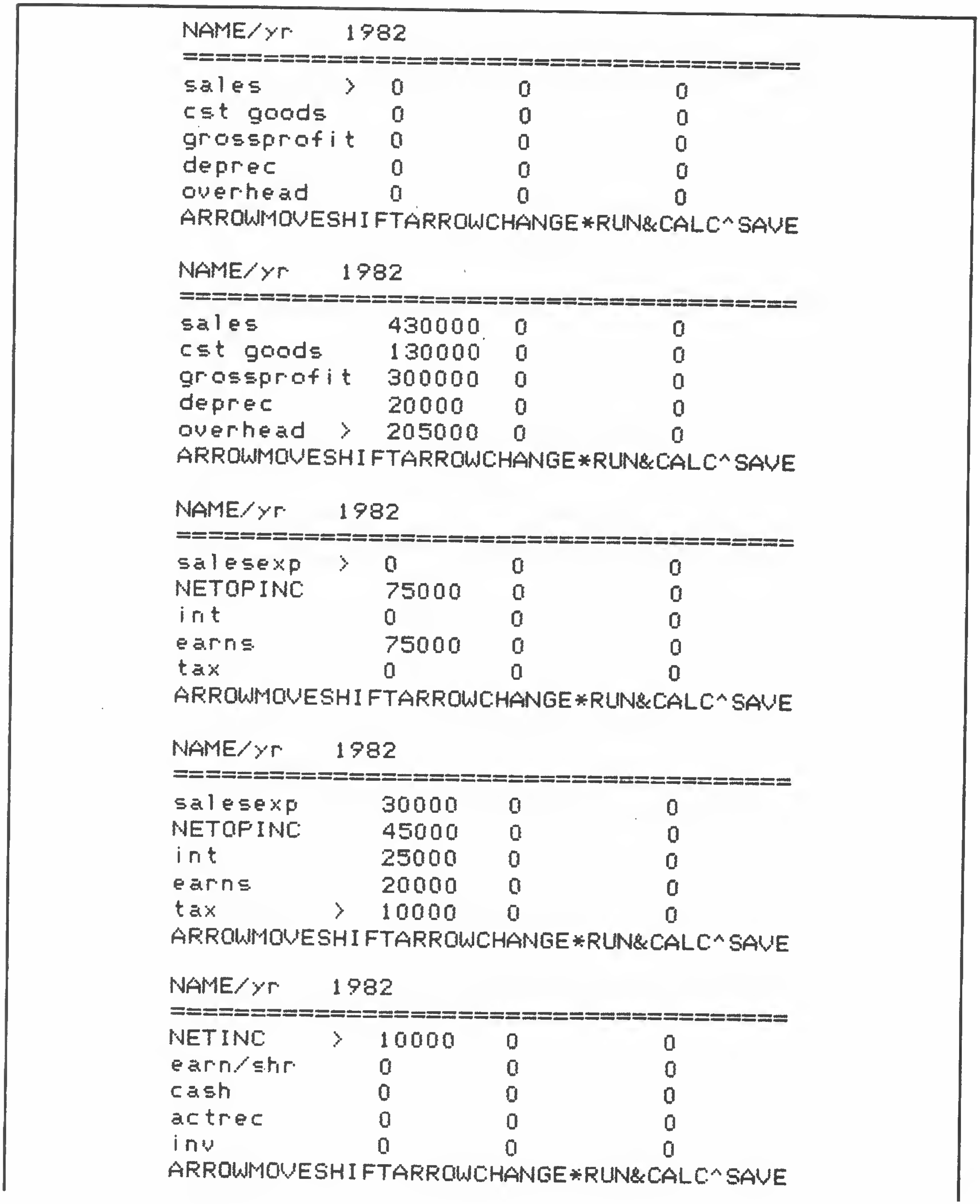
```
NAME/yr    1979        1980        1981
=======================================
totcurasts> 230000     265000      280000
fxdasts     0          0           0
TOTASTS     230000     265000      280000
actpay      0          0           0
msicacruls  0          0           0
ARROWMOVESHIFTARROWCHANGE*RUN&CALC^SAVE

NAME/yr    1979        1980        1981
=======================================
totcurasts  230000     265000      280000
fxdasts     345000     345000      400000
TOTASTS     575000     610000      680000
actpay      120000     145000      150000
msicacruls> 0          0           0
ARROWMOVESHIFTARROWCHANGE*RUN&CALC^SAVE
```



```
NAME/yr     1979        1980        1981
========================================
TOTCURLIB >   120000    145000      150000
longterm      0         0           0
cmnstock      0         0           0
pdincap       0         0           0
retearn       0         0           0
ARROWMOVESHIFTARROWCHANGE*RUN&CALC^SAVE

NAME/yr     1979        1980        1981
========================================
TOTCURLIB     120000    145000      150000
longterm      200000    200000      300000
cmnstock      500000    500000      500000
pdincap       400000    400000      450000
retearn   >   9000      10000       9000
ARROWMOVESHIFTARROWCHANGE*RUN&CALC^SAVE

NAME/yr     1979        1980        1981
========================================
retearn   >   9000      10000       9000
TOTLIB&NET    1229000    1255000    1409000
leasepay      0         0           0
debtpay       0         0           0
price         0         0           0
ARROWMOVESHIFTARROWCHANGE*RUN&CALC^SAVE

NAME/yr     1979        1980        1981
========================================
retearn       9000      10000       9000
TOTLIB&NET    1229000    1255000    1409000
leasepay      100000    100000      120000
debtpay       80000     80000       90000
price     >   10        12          12
ARROWMOVESHIFTARROWCHANGE*RUN&CALC^SAVE
```

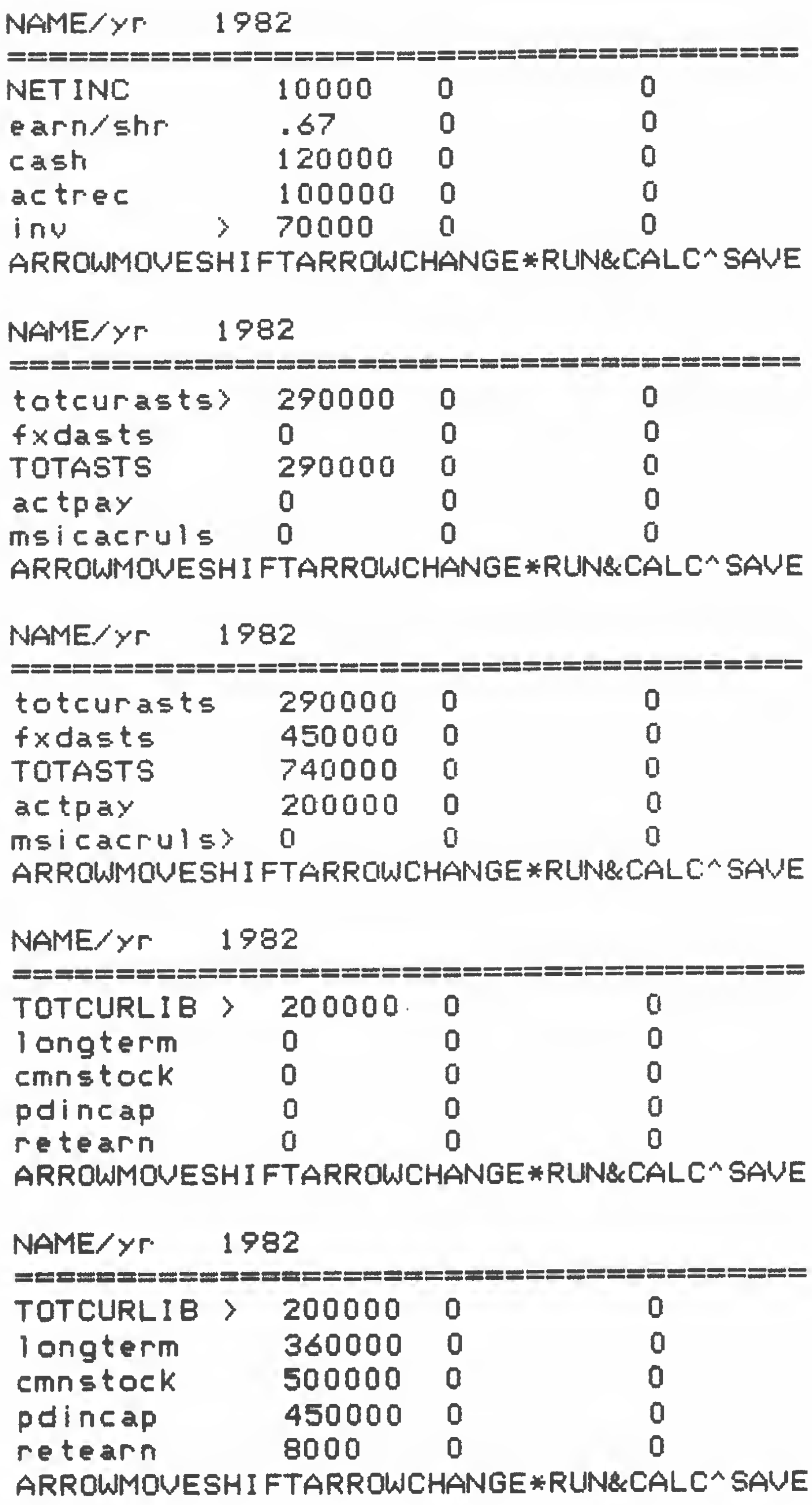
Fig. 6-7. Completing the data entry.

plot program will load the data, scale the data, and then plot the data on the display. You tell the computer to move to a new plot by pressing any key. PLOT.BA is shown in Listing 6-2.

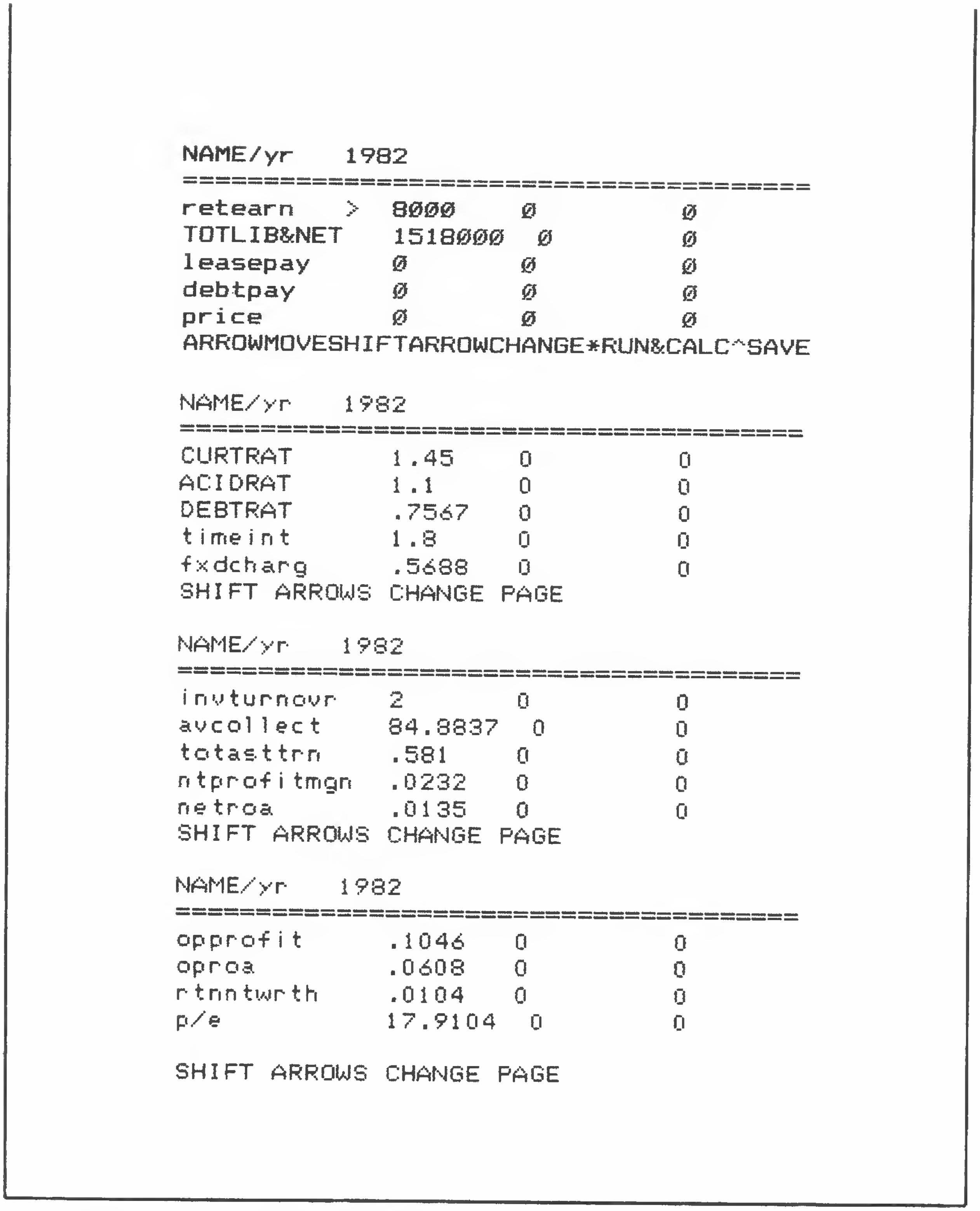


```
NAME/yr    1982
=========================================
sales      >  0         0          0
cst goods     0         0          0
grossprofit   0         0          0
deprec        0         0          0
overhead      0         0          0
ARROWMOVESHIFTARROWCHANGE*RUN&CALC^SAVE

NAME/yr    1982
=========================================
sales         430000    0          0
cst goods     130000    0          0
grossprofit   300000    0          0
deprec        20000     0          0
overhead   >  205000    0          0
ARROWMOVESHIFTARROWCHANGE*RUN&CALC^SAVE

NAME/yr    1982
=========================================
salesexp   >  0         0          0
NETOPINC      75000     0          0
int           0         0          0
earns         75000     0          0
tax           0         0          0
ARROWMOVESHIFTARROWCHANGE*RUN&CALC^SAVE

NAME/yr    1982
=========================================
salesexp      30000     0          0
NETOPINC      45000     0          0
int           25000     0          0
earns         20000     0          0
tax        >  10000     0          0
ARROWMOVESHIFTARROWCHANGE*RUN&CALC^SAVE

NAME/yr    1982
=========================================
NETINC     >  10000     0          0
earn/shr      0         0          0
cash          0         0          0
actrec        0         0          0
inv           0         0          0
ARROWMOVESHIFTARROWCHANGE*RUN&CALC^SAVE
```

Fig. 6-8. Entering data for 1982.

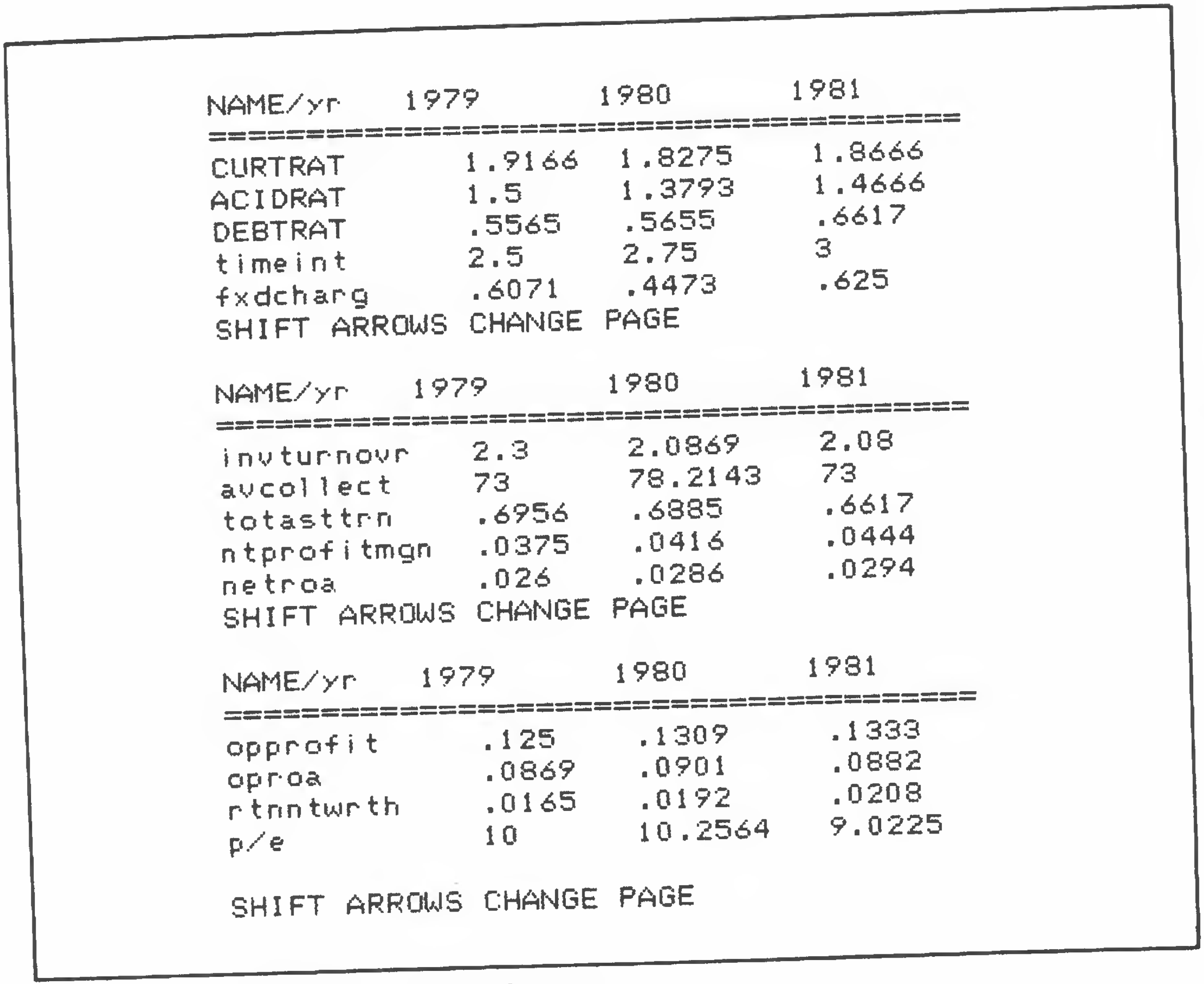


```
NAME/yr    1982
========================================
NETINC         10000    0          0
earn/shr       .67      0          0
cash           120000   0          0
actrec         100000   0          0
inv        >   70000    0          0
ARROWMOVESHIFTARROWCHANGE*RUN&CALC^SAVE

NAME/yr    1982
========================================
totcurasts>    290000   0          0
fxdasts        0        0          0
TOTASTS        290000   0          0
actpay         0        0          0
msicacruls     0        0          0
ARROWMOVESHIFTARROWCHANGE*RUN&CALC^SAVE

NAME/yr    1982
========================================
totcurasts     290000   0          0
fxdasts        450000   0          0
TOTASTS        740000   0          0
actpay         200000   0          0
msicacruls>    0        0          0
ARROWMOVESHIFTARROWCHANGE*RUN&CALC^SAVE

NAME/yr    1982
========================================
TOTCURLIB >    200000   0          0
longterm       0        0          0
cmnstock       0        0          0
pdincap        0        0          0
retearn        0        0          0
ARROWMOVESHIFTARROWCHANGE*RUN&CALC^SAVE

NAME/yr    1982
========================================
TOTCURLIB >    200000   0          0
longterm       360000   0          0
cmnstock       500000   0          0
pdincap        450000   0          0
retearn        8000     0          0
ARROWMOVESHIFTARROWCHANGE*RUN&CALC^SAVE
```



```
NAME/yr    1982
========================================
retearn    >  8ØØØ     Ø          Ø
TOTLIB&NET    1518ØØØ  Ø          Ø
leasepay      Ø        Ø          Ø
debtpay       Ø        Ø          Ø
price         Ø        Ø          Ø
ARROWMOVESHIFTARROWCHANGE*RUN&CALC^SAVE

NAME/yr    1982
========================================
CURTRAT       1.45     0          0
ACIDRAT       1.1      0          0
DEBTRAT       .7567    0          0
timeint       1.8      0          0
fxdcharg      .5688    0          0
SHIFT ARROWS CHANGE PAGE

NAME/yr    1982
========================================
invturnovr    2        0          0
avcollect     84.8837   0         0
totasttrn     .581     0          0
ntprofitmgn   .0232    0          0
netroa        .0135    0          0
SHIFT ARROWS CHANGE PAGE

NAME/yr    1982
========================================
opprofit      .1046    0          0
oproa         .0608    0          0
rtnntwrth     .0104    0          0
p/e           17.9104   0         0

SHIFT ARROWS CHANGE PAGE
```

Fig. 6-8. Entering data for 1982 (cont'd from page 87.)



```
NAME/yr    1979        1980        1981
=======================================
CURTRAT        1.9166  1.8275     1.8666
ACIDRAT        1.5     1.3793     1.4666
DEBTRAT        .5565   .5655      .6617
timeint        2.5     2.75       3
fxdcharg       .6071   .4473      .625
SHIFT ARROWS CHANGE PAGE

NAME/yr    1979        1980        1981
=======================================
invturnovr     2.3     2.0869     2.08
avcollect      73      78.2143    73
totasttrn      .6956   .6885      .6617
ntprofitmgn    .0375   .0416      .0444
netroa         .026    .0286      .0294
SHIFT ARROWS CHANGE PAGE

NAME/yr    1979        1980        1981
=======================================
opprofit       .125    .1309      .1333
oproa          .0869   .0901      .0882
rtnntwrth      .0165   .0192      .0208
p/e            10      10.2564    9.0225

SHIFT ARROWS CHANGE PAGE
```

Fig. 6-9. Results of ratio analysis for XY Widget Company.

Listing 6-1. Ratio analysis program RATIO.BA.

```
10 REM ratio calculator version 1.0
11 REM by L. E. Sparks for TAB Books
12 CLS
24 DEFSNGA-Z
25 DEFINT J,N,K
30 DIM R(8,13),X(8,28),Y$(8),C$(28),R$(13)
35 S$=STRING$(39,"=")
40 CLS
45 GOSUB10000
51 A$="1234567890.+-"
52 AZ$=CHR$(13)+CHR$(31)+CHR$(30)+CHR$(28)+CHR
        $(29)+CHR$(42)+CHR$(1)+CHR$(6)+CHR$(2)+
        CHR$(20)+"&"+"^":
```

```
     AQ$=CHR$(2)+CHR$(20)+CHR$(1)+CHR$(6)
 53 J1=0:
     J2 = 4
 54 JS=2
 55 J9=1
 56 JR=J9*10
 60 I1=0
 62 CLS:
     PRINT"DO YOU WANT TO READ DATA FROM FILE"


 64 Y$=INPUT$(1):
     IFY$="Y"THEN GOSUB9900ELSE70
 66 GOTO 90
 70 CLS:
     INPUT"ENTER NUMBER OF PERIODS ";NY
 80 FORJ=0TONY-1:
     LINEINPUT"ENTER NAME OF PERIOD ";Y$(J):
     NEXTJ
 90 GOSUB9000
 100 REM NOW CALCULATE RATIOS
 101 CLS:
     PRINT"working on ratios please wait."
 110 FORJ1=0TONY-1:
     R(J1,0)=X(J1,15)/X(J1,20):
     R(J1,1)=(X(J1,15)-X(J1,14))/X(J1,20)
 120 R(J1,2)=(X(J1,20)+X(J1,21))/X(J1,17)
 130 R(J1,3)=(X(J1,8)+X(J1,7))/X(J1,7)
 140 R(J1,4)=(X(J1,8)+X(J1,3)+X(J1,7)+X(J1,26))
         /(X(J1,7)+X(J1,26)+X(J1,27)/(1-X(J1,9)/
         X(J1,8)))
 150 IFJ1=0THENR(J1,5)=X(J1,1)/X(J1,14)ELSER(J1
         ,5)=X(J1,1)*2/(X(J1-1,14)+X(J1,14))
 160 R(J1,6)=365*X(J1,13)/X(J1,0)
 170 R(J1,7)=X(J1,0)/X(J1,17):
     R(J1,8)=X(J1,10)/X(J1,0):
     R(J1,9)=X(J1,10)/X(J1,17)
 180 R(J1,10)=X(J1,6)/X(J1,0):
     R(J1,11)=X(J1,6)/X(J1,17):
     R(J1,12)=X(J1,10)/(X(J1,22)+X(J1,23)+X(J1,
         24))
 190 R(J1,13)=X(J1,28)/X(J1,11):
     NEXTJ1
 200 REM now read in titles and then print
 210 FORJ=0TO13:
    READR$(J):
    NEXTJ
220 CLS:
    J1=0:
    J2=4
300 REM print out
310 CLS:
    PRINT "NAME/yr";TAB(10)Y$(I1);TAB(20)Y$(I1
       +1);TAB(30);Y$(I1+2)
320 PRINT STRING$(39,"=")
322 IFJ1<0THENJ1=0:
    J2=4
328 IF J2>=13THENJ2=13:

340 FOR J=J1TOJ2
345 R(I1,J)=INT(10000*R(I1,J))/10000:
    R(I1+1,J)=INT(10000*R(I1+1,J))/10000:
    R(I1+2,J)=INT(R(I1+2,J)*10000)/10000
350  PRINTR$(J);TAB(12);R(I1,J);TAB(20);R(I1+1
       ,J);TAB(30)R(I1+2,J)
360 NEXT J
370 PRINT@280,"SHIFT ARROWS CHANGE PAGE";
380 Y$=INKEY$:
    IFY$=""THEN 380
385 IFY$="P"OR Y$="p"THENGOSUB500:
    GOTO310
386 IFY$="S" ORY$="s"THEN GOSUB800:
    GOTO310
390 JZ=INSTR(AQ$,Y$):
    IFJZ=0THEN380
400 ON JZ GOTO410,430,440,450
410 J1 = J2+1:
    J2=J1+4
411 IF J1>13 THEN J1=0:
    J2=4
412 GOTO 310
430 IF J1 = 1 THEN 310
431 J2 = J1-1:
    J1 = J2 - 4
433 GOTO 310
440 IFI1<=0THENI1=0:
    GOTO300
442 I1=I1-3:
    GOTO300
450 IFI1>NYTHENI1=0:
     GOTO300
452  I1=I1+3:
     IFI1>NY-1THENI1=0:
     GOTO300
455  GOTO300
500  CLS:
     PRINT"TO PLOT DATA FIRST SAVE IT."
510  GOSUB 800
520  CLS:

530  PRINT"FILE IS NOW SAVED. GET READY TO PLOT
         ."
540  RUN"P1
800  REM write to file
810  PRINT"save to file"
820  PRINT"type of file C for cassette"
830  PRINT "R for Ram? (press R or C)"
840  Y$=INPUT$(1):
     IF INSTR("CcRr",Y$)=0THEN840
850  IF INSTR("CcRr",Y$)>2THEN A$="RAM:
     "ELSE A$="CAS:
     "
860  INPUT"enter file name ";F$
870  IF A$="RAM:
     "THENF$=A$+F$+".DO"ELSEF$=A$+F$
880  OPEN F$ FOR OUTPUT AS 1
890  PRINT"WRITING FILE."
900  PRINT#1,NY
905  FORJ=0TONY-1:
     PRINT#1,VAL(Y$(J)):
     NEXTJ
910  FOR J=0TO13:
     PRINT#1,R$(J):
     PRINTR$(J)
920  FORJ1=0TONY-1:
     PRINT#1,R(J1,J):
     PRINTR(J1,J):
     NEXTJ1
930  NEXTJ
940  CLOSE:
     RETURN
9000 REM DATA INPUT
9001 J1=0:
     J2=4
9010 CLS:
    PRINT "NAME/yr";TAB(10)Y$(I1);TAB(20)Y$(I1
       +1);TAB(30);Y$(I1+2)
9020 PRINT STRING$(39,"=")
9022 IFJ1<0THENJ1=0:
    J2=4
9028 IF J2>NTHENJ2=N:
    J1=N-4
9029 IFCC=-1THEN9035
9031 IFJS=3THEN9035ELSE9040
9035  FORJQ=0TONY
9036 X(JQ,2)=X(JQ,0)-X(JQ,1)
9037 X(JQ,6)=X(JQ,2)-X(JQ,3)-X(JQ,4)-X(JQ,5):
    X(JQ,8)=X(JQ,6)-X(JQ,7):
    X(JQ,10)=X(JQ,8)-X(JQ,9):
    X(JQ,15)=X(JQ,12)+X(JQ,13)+X(JQ,14):
    X(JQ,17)=X(JQ,15)+X(JQ,16)
9038 X(JQ,20)=X(JQ,18)+X(JQ,19):
    X(JQ,25)=X(JQ,20)+X(JQ,21)+X(JQ,22)+X(JQ,2
       3)+X(JQ,24):
    NEXTJQ
9040 FOR J=J1TOJ2
9050  PRINTC$(J);TAB(12);X(I1,J);TAB(20);X(I1+
       1,J);TAB(30)X(I1+2,J)
9060 NEXT J
9070 PRINT @(JS*40)+JR,">";
9075 PRINT@280,"ARROWMOVESHIFTARROWCHANGE*RUN&
       CALC^SAVE";
9080 Y$=INKEY$:
    IFY$=""THEN 9080
9085 IFY$="&"THENCC=-1:
    GOTO9010
9086 CC=0
9090 IFINSTR(A$,Y$)<>0 THEN 9500
9100 JZ=INSTR(AZ$,Y$):
    IFJZ=0THEN9080
9105 REM PRINTY$,JZ:
    STOP
9110 ON JZ GOTO 9120,9120,9140,9160,9180,9199,
       9300,9350,9200,9250,9250,9800
9120 PRINT@(JS*40)+JR," ";
9122 JS=JS+1:
    IFJS>6THEN JS=2:

9124 PRINT@(JS*40)+JR,">";
9130 GOTO 9080
```

```
9140 PRINT@(40*JS)+JR," ";
9150 JS =JS-1:
    IFJS<2 THEN JS=6
9152 PRINT@(40*JS)+JR,">";:
    GOTO9080
9160 PRINT@(JS*40)+JR," ";
9170 J9=J9+1:
    IFJ9>I1+3THEN9176ELSEJR=J9*10
9172 IF I1+J9-2>NYTHENJ9=1:
    JR=J9*10
9173 IFJ9+I1>NYTHENJ9=1:
    JR=J9*10
9175 PRINT@(JS*40)+JR,">";:
    GOTO 9080
9176 J9=1:
    JR=J9*10:
    GOTO9010
9179 GOTO9010
9180 PRINT@(JS*40)+JR," ";
9182 J9=J9-1:
    IFJ9<1THEN9192ELSEJR=J9*10
9190 PRINT@(JS*40)+JR,">";:
    GOTO 9080
9192 J9=1:
    JR=10*J9
9194 REM
9195 J9=1:
    JR=10*J9:
    GOTO9000
9199 RETURN
9200 CC=-1:
     J1 = J2+1:
    J2=J1+4
9210 JR =J9*10:
     JS=2
9220 GOTO 9010
9250 IF J1 = 1 THEN 9010
9260 CC=-1:
    J2 = J1-1:
    J1 = J2 - 4
9270 GOTO 9010
9300 CC=-1:
    IFI1=0THENI1=0:
    J9=1:
    JR=J9*10:
```

```
    GOTO9000
9310 I1=I1-3:
    J9=1:
    JR=J9*10:
    GOTO9000
9350 CC=-1:
     IFI1>NYTHENI1=0:
    J9=1:
    JR=J9*10:
    GOTO9000
9360 I1=I1+3:
    J9=1:
    JR=J9*10:
    IFI1>NY-1THENI1=0:
    GOTO9000
9365 GOTO9000
9500 PRINT @(40*JS)+JR,"  ";
9505 PRINT@280,"ENTER REST OF DATA PRESSENTER
       WHEN DONE";
9510 V$=Y$
9530 PRINT@(40*JS)+JR,V$;CHR$(95);"  ";
9540 Y$=INKEY$:
    IF Y$ =""THEN 9540
9550 IF Y$=CHR$(13) THEN 9700
9560 IF Y$<>CHR$(29) THEN 9600
9570 IF LEN (V$)=0 THEN 9530
9580 V$=LEFT$(V$,LEN(V$)-1):

9590 Y$="":
    GOTO 9530
9600 IF INSTR(A$,Y$)=0 THEN 9530
9610 V$ = V$ + Y$
9620 GOTO 9530
9700 X(I1+J9-1,J1+JS-2)=VAL(V$)
9702 J9=J9+1:
    IFJ9>3THENJ9=1
9703 IFJ9+I1>NYTHENJ9=1
9704 JR=J9*10
9730 GOTO 9010
9800 REM SAVE TO FILE
9810 CLS:
    PRINT"STORE DATA TO FILE"
9820 INPUT"ENTER NAME OF FILE ";F$
9830 PRINT"PRESS C FOR CASSETTE OR R FOR RAM"
```

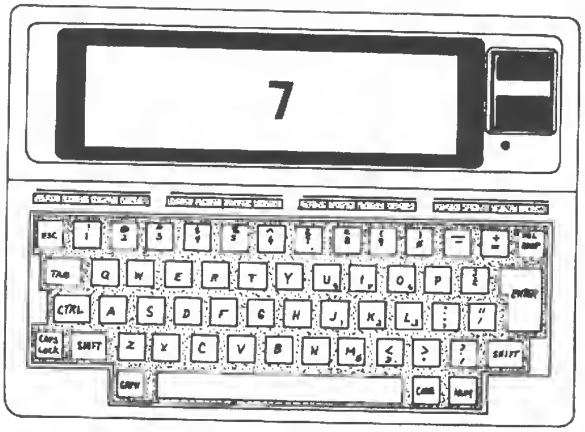
```
9840 Y$=INPUT$(1):
    IFY$="R" THEN F$="RAM:
    "+F$+".DO" ELSE F$="CAS:
    "+F$
9850 OPEN F$ FOROUTPUT AS 1
9860 PRINT#1,NY
9870 FORJ=0TONY-1:
    PRINT#1,Y$(J):
    NEXTJ
9880 FORJ1=0TON:
    FORJ=0TONY-1:
    PRINT#1,X(J,J1):
    NEXTJ:
    NEXTJ1
9890 CLOSE:
    J1=0:
    J2=4:
     GOTO 9010
9900 REM ROUTINE TO READ IN DATA FROM FILE
9910 CLS:
    PRINT"GET DATA FROM FILE"
9920 INPUT"ENTER FILE NAME";F$:
    PRINT"PRESS R FOR RAM FILE C FOR CASSETTE"
        :
    Y$=INPUT$(1):
    IFY$="R"THENF$="RAM:
    "+F$+".DO"ELSEF$="CAS:
    "+F$
9925 OPENF$FORINPUTAS1
9930 INPUT#1,NY:
    FORJ=0TONY-1:
    INPUT#1,Y$(J):
    NEXTJ:
    FORJ1=0TON:
    FORJ=0TONY-1:
    INPUT#1,X(J,J1):
    NEXTJ:
    NEXTJ1:
    CLOSE
9940 PRINT"DO YOU WANT TO ADD ADDITIONAL DATA?
        "
9950 Y$=INPUT$(1):
    IFY$<>"Y"THEN RETURN
9960 CLS:
     INPUT"ENTER NUMBER OF ADDITIONAL YEARS ";N
        N
9970 FOR J=NYTONY+NN-1:
     INPUT"NAME OF PERIOD ";Y$(J):
     PRINTY$(J):
     NEXTJ:
     NY=NY+NN-1:
     RETURN
10000 REM set up
10010 J=0
10020 READ T$:
     IFT$="end"THENN=J-1:
     RETURN
10030 C$(J)=T$:
     J=J+1:
     GOTO10020
12000 REM write to file
12010 PRINT"save to file"
12020 PRINT"type of file C for cassette"
12030 PRINT "R for Ram? (press R or C)"
12040 Y$=INPUT$(1):
     IF INSTR("CcRr",Y$)=0THEN1040
12050 IF INSTR("CcRr",Y$)>2THEN A$="RAM:
     "ELSE A$="CAS:
     "
12060 INPUT"enter file name ";F$
12070 IF A$="RAM:
     "THENF$=A$+F$+".DO"ELSEF$=A$+F$
20000 DATA sales,cst goods,grossprofit,deprec,
        overhead,salesexp,NETOPINC,int,earns,ta
        x,NETINC,earn/shr,cash,actrec,inv,totcu
        rasts,fxdasts,TOTASTS,actpay,msicacruls
        ,TOTCURLIB,longterm,cmnstock,pdincap,re
        tearn,TOTLIB&NET
20001 DATA leasepay,debtpay,price,end
20002 DATA CURTRAT,ACIDRAT,DEBTRAT,timeint,fxd
        charg,invturnovr,avcollect,totasttrn,nt
        profitmgn,netroa,opprofit,oproa,rtnntwr
        th,p/e
```

Listing 6-2. Plotting program PLOT.BA used with RATIO.BA.

```
1 REM PLOT PROGRAM FOR RATIO ANALYSIS
2 REM VERSION 1 BY L. E. SPARKS
```

```
3 REM FOR TRS-80 MODEL 100
4 REM
10 CLS
20 DIM X(10),Y(10)
22 INPUT"FILE NAME ";F$:
    F$="RAM:
    "+F$+".DO":
    OPENF$FORINPUTAS1:
    INPUT#1,NY:
    FORJ=0TONY-1:
    INPUT#1,X(J):
    PRINTX(J):
    NEXTJ
30 CLS
32 IFEOF(1)THENCLOSE:
    END
34 INPUT#1,C$:
    FORJ=0TONY-1:
    INPUT#1,Y(J):
    PRINTY(J):
    NEXTJ:

40 XI=1E9:
    XM=-1E9:
    YM=XM:
    YI=XI
50 FORJ=0TONY-1
60 IFX(J)>XMTHENXM=X(J)
70 IFX(J)<XITHENXI=X(J)
80 IFY(J)>YMTHENYM=Y(J)
90 IF Y(J)<YITHENYI=Y(J)
100 NEXTJ
115 IF XM=XITHENXM=2*XI:
    XI=.5*XI
116 IF YI=YMTHENYM=2*YI:
    YI=.5*YI
120 DX=XM-XI:
    DY=YM-YI:
    SX=180/DX:
    SY=55/DY
130 PRINTDX,SX,DY,SY
140 REM scale
145 CLS
150 PRINT@284,XI;:
    PRINT@313,XM;
```

```
240 PRINT@240,;:
    PRINTUSING"###.##";YI;:
    PRINT@0,;:
    PRINTUSING"###.##";YM;
245 LINE(34,56)-(234,56)
246 LINE (34,56)-(34,0)
247 FOR J=0TONY-1:
    YP=Y(J)-YI:
    YP=YP*SY:
    YP=56-YP:
    XP=X(J)-XI:
    XP=34+XP*SX
248 IF J=0THENX1=XP:
    Y1=YP:
    GOTO252
249 LINE(X1,Y1)-(XP,YP)
250 X1=XP:
    Y1=YP
252 NEXTJ
260 X=33:
    LINE(X,55)-(X,56):
    FORJ=1TONY-1:
    X=X+SX:
    LINE(X,55)-(X,57):
    NEXTJ
270 PRINT@80,C$;
280 Y$=INKEY$:
    IFY$=""THEN280
290 GOTO 30
```



Forecasting

All financial decisions require forecasting the future. In the final analysis, the quality of the decision often depends on the quality of the forecast. The forecast provides the raw data for the quantitative financial analyses discussed in most of this book. The expected growth in income and the expected escalation in expenses are the end result of forecasts, and are the essential input data for the analysis process. Forecasting therefore deserves very careful attention.

One of the major advantages of computerized financial decision making is that you can analyze different futures and see how various decisions fare under different conditions. Even if you don't always make the best decision, you often can avoid the worst one.

The type of forecast you need depends on the problem. In some cases, such as forecasts of inflation rates, you can find reasonable forecasts published in many places. Thus, there are some factors that you may not need to forecast personally. You can depend on the experts. This does not mean that you should always depend on the experts, nor that with a little work you can't do better than (or at least as well as) the experts.

However, when you get to the specifics of your own company or your own financial well-being, you are likely to need to do your own forecasting. You will quite likely find that the expert forecasts are ideal for the country as a whole, but not for your particular unique situation. For example, you may find forecasts for retail sales nationwide but not for your state or city, and definitely not for your store. If you want a forecast for your sales, you'll have to do it yourself—and you will find that you can forecast your own situation better than the experts, if you put in the necessary work. Your edge is your knowledge of your business.

TYPES OF FORECASTS

There are two basic types of forecasts, *qualitative* (or *technological*) and *quantitative*. In general, qualitative forecasts deal with the long term—next year or the next ten years. Quantitative forecasts deal with the short term—next month or next quarter or perhaps even next year.

Qualitative forecasts are often used in an attempt to determine when new trends will replace old trends and when new technologies will replace old ones. A major emphasis of qualitative forecasting is prediction of how the future will differ from the past. Such forecasts are expensive to prepare and are reserved for important projects and large companies. In fact, governments are often the only organizations that can afford to use qualitative forecasts as a regular tool.

The reason that qualitative forecasts are expensive is that they are based on expert opinion—and of course experts are expensive. Several tools have been developed to increase the productivity of the expert panels used to make qualitative forecasts. These tools are all designed to obtain some type of consensus from the experts in as short a time as possible.

Individuals and small companies can make qualitative forecasts, but they are generally based on the opinions of one or two people and not a large panel of experts. Thus, we generally have to depend on the published qualitative forecasts of the experts—especially those paid for by governments. Although a computer can be used to tabulate the experts' opinions, it is of little use in actually preparing a qualitative forecast.

Quantitative forecasts are based on the assumption that the future will follow the past and that the key to the future is contained in the data from the past. In general, quantitative forecasts have had a fairly good record for forecasting short-term futures. For example, next month's sales and next quarter's earnings can generally be forecast. Next year's sales and earnings are much more difficult to predict.

There are two approaches to finding the key to the future from past data. The first approach, called *causal forecasting,* is based on the assumption that there are causal relationships between various variables which account for the past behavior. Thus, the study of the past data is devoted to building a mathematical model of the past. The model is then used to predict the future. Models of individual companies, groups of companies, and whole countries have been built and used with various degrees of success.

Regression analysis, one of the main tools of causal modeling, is a mathematical procedure designed to find the best relationship between variables. One of the variables is called the *dependent variable* because it depends on the other variables, called the *independent variables.* A dependent variable can be a function of several independent variables.

A model based on regression analysis can be as simple as a single equation relating one dependent variable to one independent variable, or as complex as some of the econometric models having hundreds of equations relating hundreds of variables.

One of the major advantages of causal models is that they allow a decision maker to see the consequences of a particular decision. For example, if you have a regression model relating sales to advertising, you can quickly determine the consequences of a particular level of advertising. Then, if you have a similar model showing the relation between sales and costs (inventory requirements, overheads, etc.), you can determine an optimum level of advertising.

The second approach to quantitative forecasting, called *time series forecasting,* is based on the assumption that past trends with time will continue into the future. No assumption is made about causal relationships between variables. All that matters is the trend with time. In other words, time series analysis assumes that there is a pattern to the historical data, and that the basic pattern will hold for the future. The problem is finding the pattern.

Time series forecasting is generally simpler than regression analysis. The price of simplicity is the loss of the ability to simulate what happens as a result of your decisions. You should note that time series forecasting can produce just as good (and just as bad) forecasts as causal forecasting. Which of the two quantitative methods, regression analysis or time series analysis, is at best is an open question. Quite often time series analysis is selected by default because of the impossibility of determining all the factors necessary for a good causal model. Often, even if all the important factors can be identified, the necessary data may not be available.

Regardless of which method of quantitative forecasting you use, you should plot the data to see what relationships may be hidden. The plot can then help you determine what kind of forecasting scheme you should use.

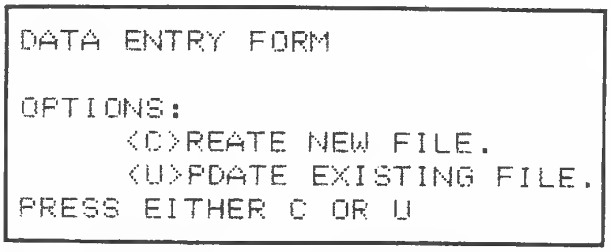
The rest of this chapter will be devoted to programs to help you prepare quantitative forecasts. The programs allow you to enter data, plot data, conduct regression analysis, and conduct time series analysis.

DATA ENTRY

All the working programs are designed to read data from RAM or cassette files. By using RAM or cassette to store data for the program, we avoid the need to write and type in separate data entry modules for each of the programs. We save time now and space in the Model 100 later.

The data entry program in Listing 7-1 is a dual-purpose program. It provides a way to create an original data file and also a way to edit or update the file. The program is built along the same lines as the matter data entry module. It works the same way and doesn't require comment now. If you have used any of the other programs in the book, you already know how the data entry program works.

The program signs on with the master menu shown in Fig. 7-1. You are asked whether you want to create a new file or update an old file. Press C to create a new file and press U to update an old file. If you elect to create a new file, you will be asked for identifying information about the particular run you are making. The information the program wants are the names of the variables and the time period covered. You also have a chance to enter remarks to identify the data for later use.



```
DATA ENTRY FORM

OPTIONS:
      <C>REATE NEW FILE.
      <U>PDATE EXISTING FILE.
PRESS EITHER C OR U
```

Fig. 7-1. Master menu for forecasting data entry.

When you have the data, you can store them by pressing the * key. You will be asked for a filename and whether the file is a RAM or a cassette file. If the file is a cassette file, you will be asked to press any file key when the recorder is ready.

If you elect to update a file, you will be asked to enter the name of the file you want to update. You are then asked if the file is a RAM or a cassette file. If it is a cassette file, the program will ask you to press any key when the recorder is ready. The program will load the identifying information from the file, display it, and then ask you if this is the file you want. If it is not, you are returned to the master menu; if it is, all the data are loaded.

As soon as the data are loaded, they are displayed in the standard data form. You can move around the form just as you did when you created the file. You can correct any entry and you can add entries at the end of the file.

When you have finished updating the file, press * to store the file. You will be asked for a filename; you may use either the old or the new filename.

The data entry and file creation program is useful enough that you may want to use it for other purposes. If so, note that the program creates files with the following structure:

R\$ a string used for the remarks
T\$ a string that includes the time period of the data
X\$ a string used for the name of the X variable
Y\$ a string used for the name of the Y variable
N the number of data points in the file
X(1), Y(1) . . . X(N), Y(N) where the data points are stored as X, Y pairs

USING THE PLOTTING PROGRAM

The first step in data analysis is plotting the data, either on graph paper or on a computer. The basic idea is to get a feel for the possible patterns that exist in the data. The type of pattern you see should be used to guide the way you proceed with your future analysis.

For example, look at the plot of sales versus advertising in Fig. 7-2. You don't have to look very hard to determine that some kind of relationship exists between the two factors. On the other hand, look at the data plotted in Fig. 7-3 showing the same relation for another business. There is certainly no strong relation between the two factors.

This example shows how a simple plot can help determine how we go about forecasting the future. In the first case, where the plot showed a strong relation between advertising and sales, we can apply regression analysis. In the second case, we would probably have better luck using time series analysis.

The program PLOT.BA, Listing 7-2, is a general plotting program that can plot data in three ways—Y versus X, Y versus time, and X versus time. The program automatically scales the plot to adjust it to the data. The program requires that the data be stored either in a RAM file or in a tape file. The general data entry module given earlier in the book can be used to create the data file for the plotting program. PLOT.BA is similar to the plotting program used to plot the results of the RATIO.BA program.

REGRESSION ANALYSIS

The most common form of regression analysis is *linear regression.* Linear regression is based on the idea that a straight line can be drawn through the plotted data points. Thus, if the plot of the data shows that a straight line is a reasonable approximation of the data, linear regression is justified.

The mathematical technique of linear regression draws the "best" straight line through the data. "Best" in this case means that the sum of the squares of the differences between the actual data and the straight line is minimized. That is, the sum of the squared differences for any other straight line through the data will be greater than the sum of the squared differences for the best straight line.

The important thing to remember is that the best straight line has definite mathematical meaning. Because of this strict mathematical meaning of the meaning of best, we can say how well the line fits the data. The parameter that tells us how well the line fits is the *regression coefficient,* often called "R squared." R squared tells us roughly what fraction of the variation in the dependent variable (Y) is explained by the regression equation. If R squared were equal to 1.0 (the maximum value), then the regression equation explains all the variation in the data. Generally if R squared is less than 0.8 or 0.9, the regression equation should be viewed with caution.

The linear regression program is presented in Listing 7-3 for LINREG.BA, which requires that the data be stored in a RAM or cassette file. The master data entry program should be used to create the file.

As an example showing the use of LINREG.BA, suppose that you run a small store and have collected the following data on your advertising expenses and sales. You figure you can handle weekly sales of about $8,000 before you need to expand—and you don't want to expand. The question then is to find the advertising expenditure that will give the desired level of sales.

In this case the sales are the dependent variable, i.e., the Y in the regression equation. Enter the data using the data entry program, remembering that data entry works the same way as in the data entry modules in the rest of the programs. You move from item to item with the up and down arrows and from page to page with the shift up and down arrows. When you are sure that the data are correct, press S to save the data. You can save the data to a RAM file or to a tape file. If the data are important, I suggest that you use the tape file. The data entry for this example is shown in Fig. 7-4.

As soon as the data are saved, press Q to exit the data entry program and then type RUN"LINREG. This will load the linear regression program. Tell the computer the name of the file to plot, then tell it whether the file is a RAM or a cassette file—and then let the computer do the work. The answer is shown in Fig. 7-5.

Note that the value of the linear constants and R squared are printed. You can now use the equation to forecast the impact of various levels of

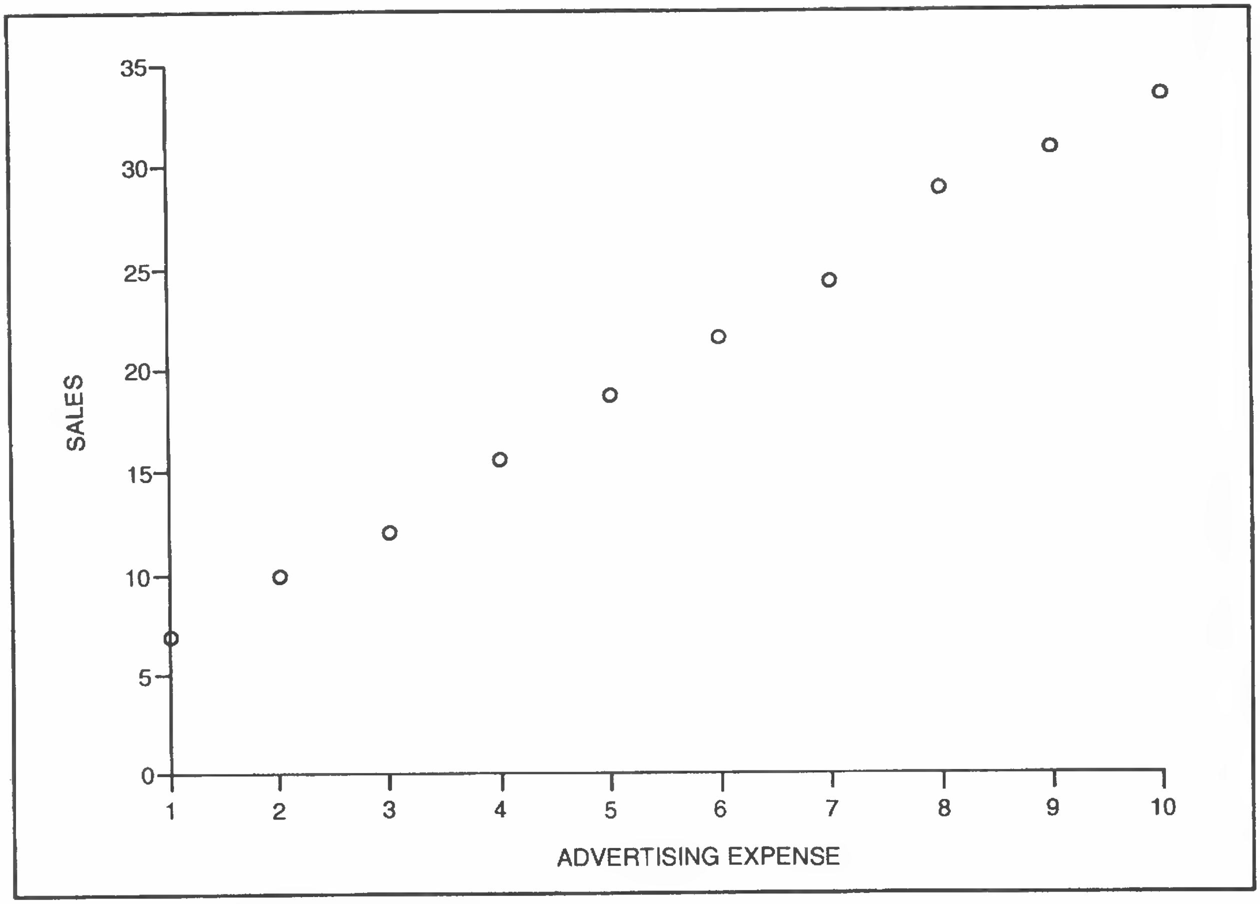


Fig. 7-2. Spotting relationships with plotted data.

spending. Try several values to see how much you need to spend to get your $8,000 a week in sales.

TIME SERIES FORECASTING

As mentioned above, the concept of time series forecasting is that the history of the data holds the key to predicting the future. The problem is to find the key.

Time series forecasting methods are the most popular of the various forecasting techniques. Considerable research has been done to determine the best ways of forecasting and to develop better forecasting techniques. Although this research has proven the utility of time series forecasting, especially for short-term forecasts, it has not uncovered the universal best forecasting method.

Time series forecasting techniques are often called *smoothing* techniques. One of the objectives of time smoothing is to remove the random "noise" from the data, leaving a smooth curve. This smooth curve is then extrapolated into the future to produce the forecast. The most common smoothing methods are *moving averages* and *exponential smoothing.*

Exponential Smoothing

Exponential smoothing is well suited for the Model 100 because it is fast, and very little data must be stored in the computer to use the technique. The formula for exponential smoothing is:

$$F(t) = F(t-1) + A \times [Y(t-1)-F(t-1)]$$

where F(t) is the forecast for period t, F(t−1) is the forecast for period t−1 (the previous time period).

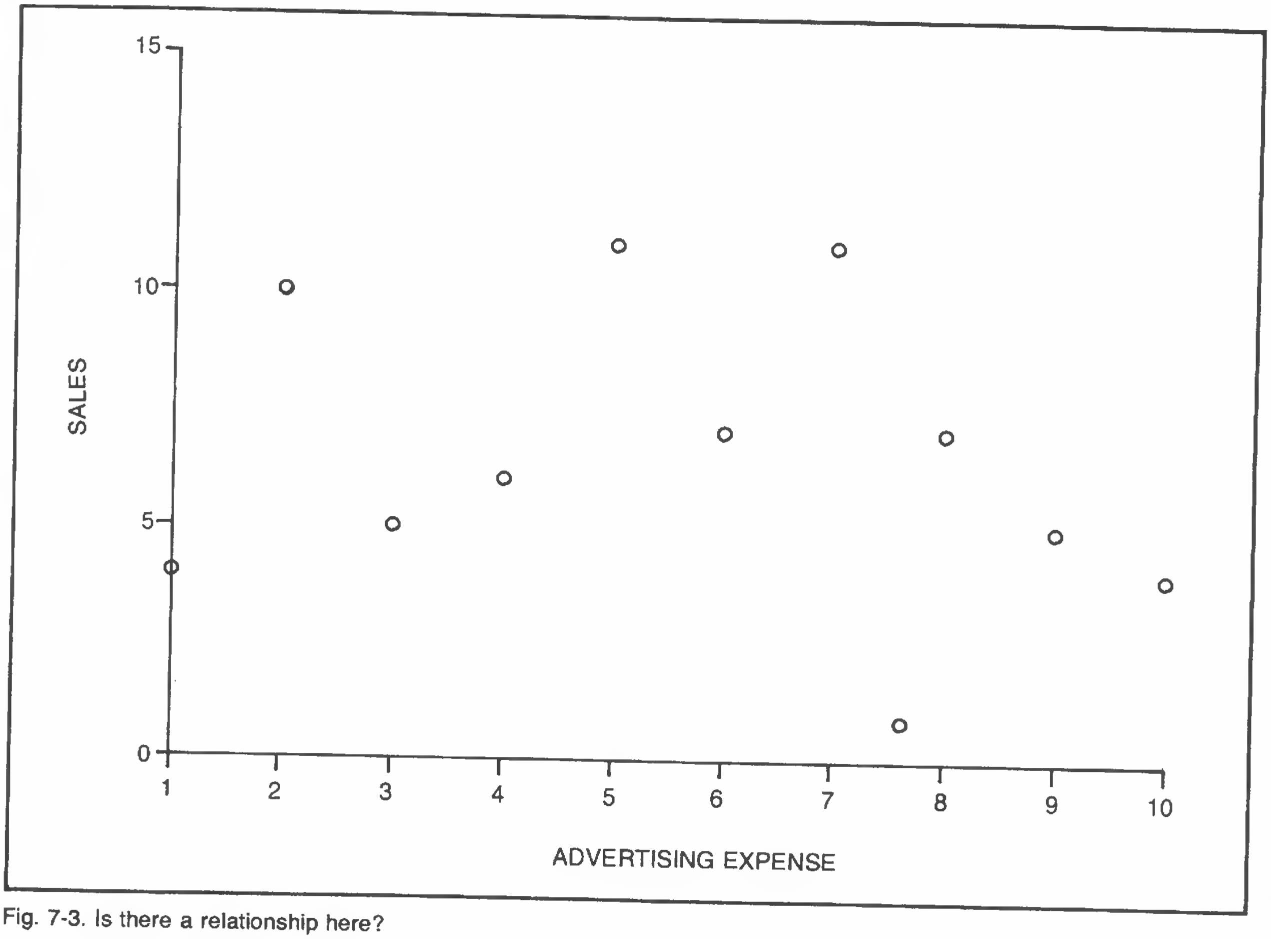


Fig. 7-3. Is there a relationship here?

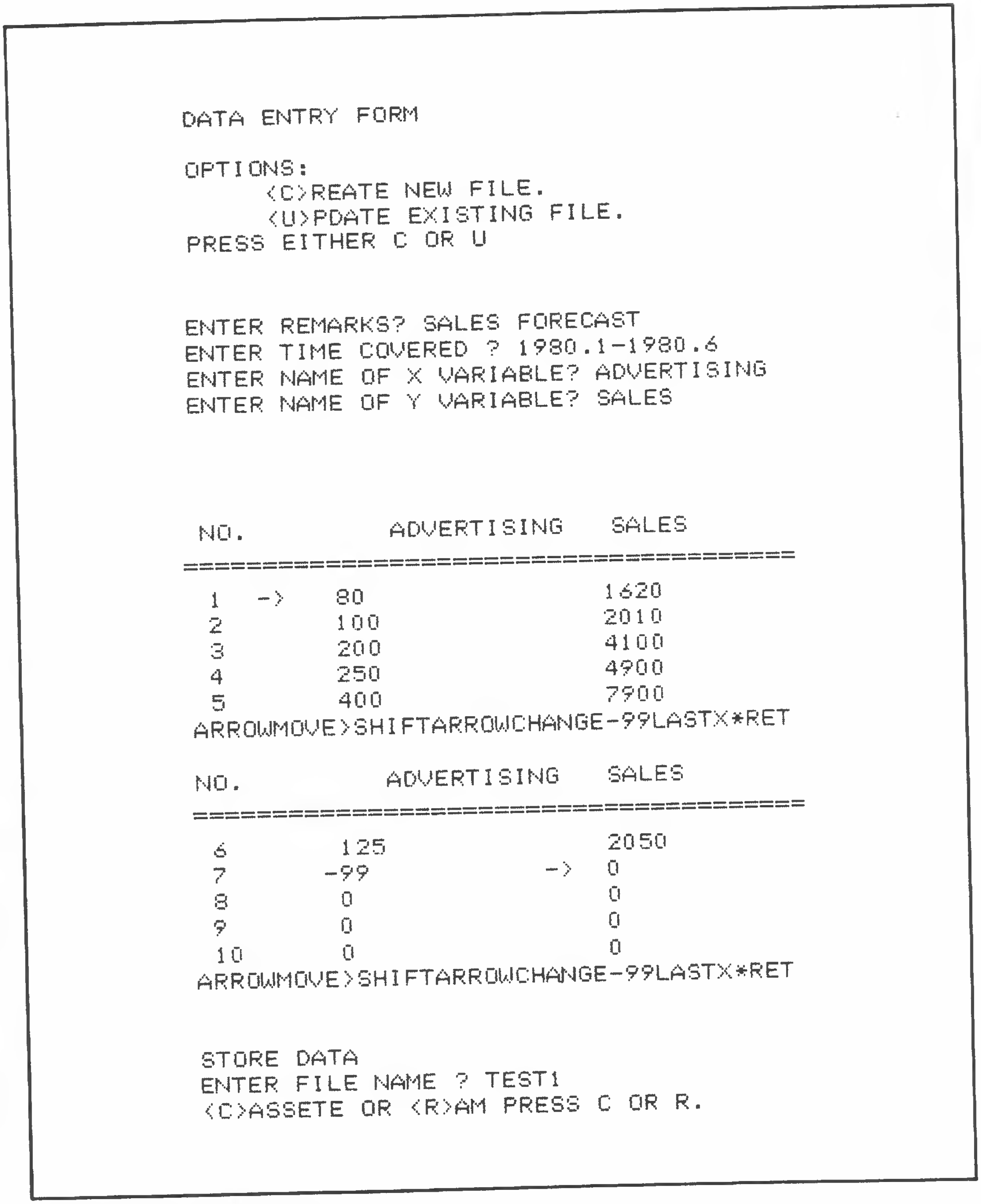
A is the smoothing parameter, and Y (t−1) the actual value of Y during the previous period. What this equation says is that our forecast for next time period is the forecast for the previous period plus a correction term. The correction term is a fraction of the error between the previous forecast and the previous actual data.

As you can see from the equation, the data required to use exponential smoothing are the smoothing parameter, the actual value of Y at t−1, and the forecast for Y at t−1. These minimal data requirements are ideally suited for the Model 100. The RAM required to store the data will almost never be missed.

The problem with exponential smoothing is determining the value of A. Maximum smoothing is achieved with small values of A; large values of A give little smoothing and are generally looked on with disfavor. Values of about 0.3 are commonly recommended.

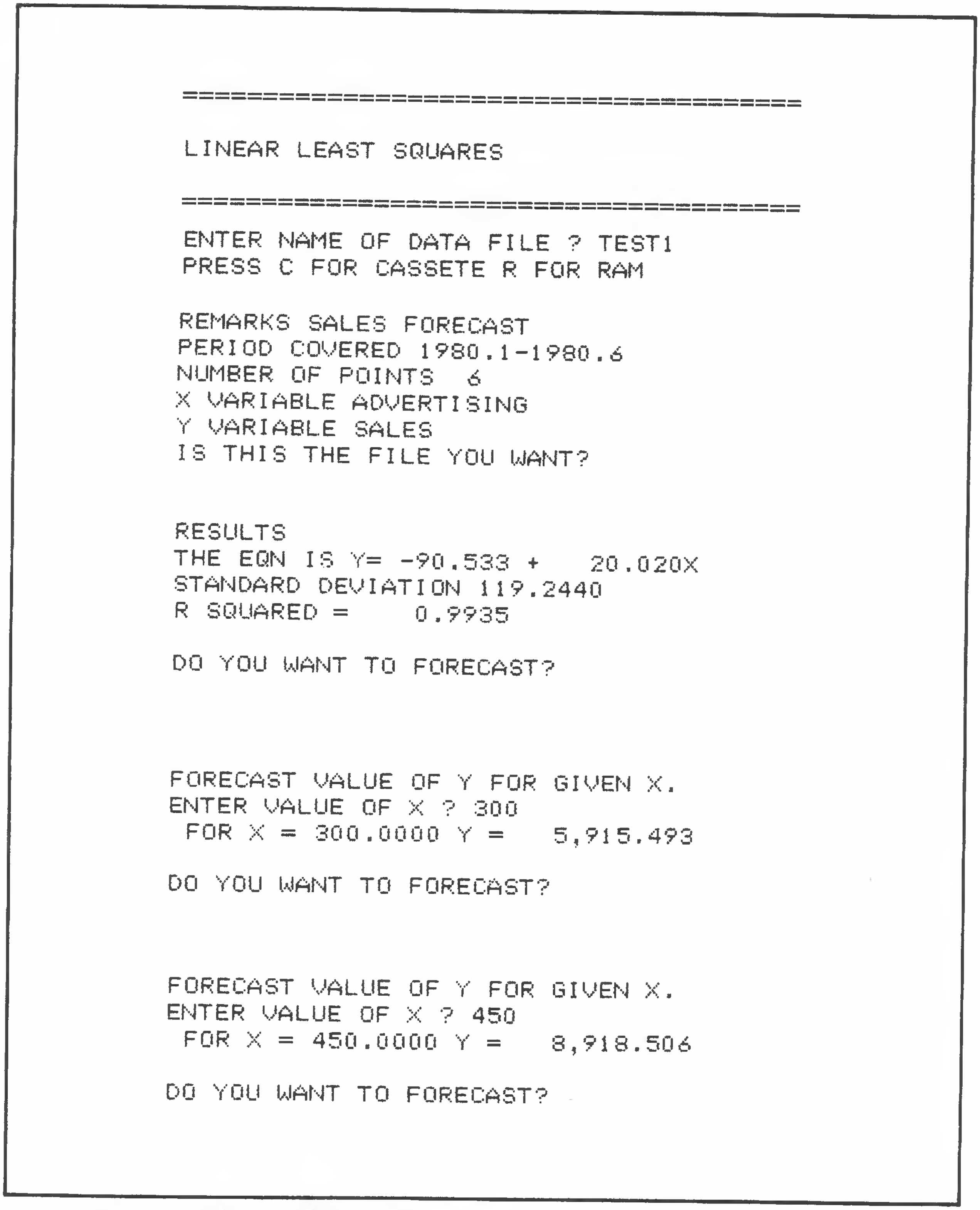
One of the major problems with all forecasting techniques is the problem of determining when the data being forecast have changed. When the data change, the forecasting environment has also changed. If the forecasting environment has changed, then our forecasts have no chance of being accurate. The program presented in Listing 7-4 provides an automatic signal that warns us that the data have changed. The program does not provide for automatic adjustment of the forecast. That's your job. What it is designed to do is perform exponential smoothing forecasting and provide you with information about how well the forecast is doing.

The program uses a smoothed error test first proposed by Trigg ("Monitoring a Forecasting System", *Operational Research Quarterly,* Volume



```
DATA ENTRY FORM

OPTIONS:
      <C>REATE NEW FILE.
      <U>PDATE EXISTING FILE.
PRESS EITHER C OR U


ENTER REMARKS? SALES FORECAST
ENTER TIME COVERED ? 1980.1-1980.6
ENTER NAME OF X VARIABLE? ADVERTISING
ENTER NAME OF Y VARIABLE? SALES



 NO.            ADVERTISING   SALES
=======================================

  1   ->    80                 1620
  2         100                2010
  3         200                4100
  4         250                4900
  5         400                7900
ARROWMOVE>SHIFTARROWCHANGE-99LASTX*RET

 NO.            ADVERTISING   SALES
=======================================

  6         125                2050
  7        -99            ->   0
  8         0                  0
  9         0                  0
  10        0                  0
ARROWMOVE>SHIFTARROWCHANGE-99LASTX*RET


 STORE DATA
 ENTER FILE NAME ? TEST1
 <C>ASSETE OR <R>AM PRESS C OR R.
```

Fig. 7-4. Data entry for LINREG.BA.



```
=======================================

LINEAR LEAST SQUARES

=======================================

ENTER NAME OF DATA FILE ? TEST1
PRESS C FOR CASSETE R FOR RAM

REMARKS SALES FORECAST
PERIOD COVERED 1980.1-1980.6
NUMBER OF POINTS  6
X VARIABLE ADVERTISING
Y VARIABLE SALES
IS THIS THE FILE YOU WANT?


RESULTS
THE EQN IS Y= -90.533 +    20.020X
STANDARD DEVIATION 119.2440
R SQUARED =     0.9935

DO YOU WANT TO FORECAST?



FORECAST VALUE OF Y FOR GIVEN X.
ENTER VALUE OF X ? 300
 FOR X = 300.0000 Y =    5,915.493

DO YOU WANT TO FORECAST?



FORECAST VALUE OF Y FOR GIVEN X.
ENTER VALUE OF X ? 450
 FOR X = 450.0000 Y =    8,918.506

DO YOU WANT TO FORECAST?
```

Fig. 7-5. Results of linear regression (least squares) exampie.

15, 1964). The error test is given by:

$$E1(t) = A \times E(t) + (1 - A) \times E1(t-1)$$

$$M(t) = A \times abs[E(t)] + (1-A) \times M(t-1)$$

$$T(t) = \frac{E1(t)}{M(t)}$$

where E1(t) is the current average error, A is the exponential smoothing constant (often 0.1 or 0.2), E(t) is the current or most recent forecasting error, abs[E(t)] is the absolute value of the current forecasting error, and M(t) is the mean absolute deviation (evaluated as the exponentially weighted average of the absolute errors) for the current period, and T(t) is the tracking signal for the current period, which can range from−1 to+1. The larger the value of the tracking signal, the more likely it is that the data have changed. When T exceeds 0.7, the forecast is generally considered to be out of control. When the forecast is so judged, the forecaster should take corrective action. Generally this means that a new forecast is prepared and then the system starts over.

This program requires slightly more information than does the simple exponential smoothing technique. The data required are the smoothing constant, the current forecasting error, the mean absolute deviation, and the average forecasting error. These are two more data items than are required for simple exponential smoothing.

The program prompts for all the information it needs. The first time you run the program you must give the program the filename for the file where the information for subsequent forecasts is to be stored. Also, when you first use the program you must supply both the actual value of the forecast variable for the first period and an estimate of the forecast for that period. What value you pick for the estimated initial forecast is up to you. Something around 10 percent or so of the actual value generally works out well. You also must enter the smoothing constant. Values between 0.2 and 0.3 are generally recommended in the forecasting literature.

After you have provided all the initial information, the program makes a forecast for the second period. The forecast, an estimate of the standard deviation of the data, and the tracking signal are printed. The data needed to make the next forecast are written to a RAM file.

To use the program to make forecasts of later time periods, first run the program, tell it the name of the RAM file, and then enter the most recent value of the forecast variable. The program will print the forecast, standard deviation, and tracking signal for the next time period. A warning message will be printed if the absolute vale of the tracking signal is greater than 0.7.

Note that this program does not use the data stored in RAM by the data entry program. The program does not require the large amount of data the other programs require. Thus it has its own, short data entry module.

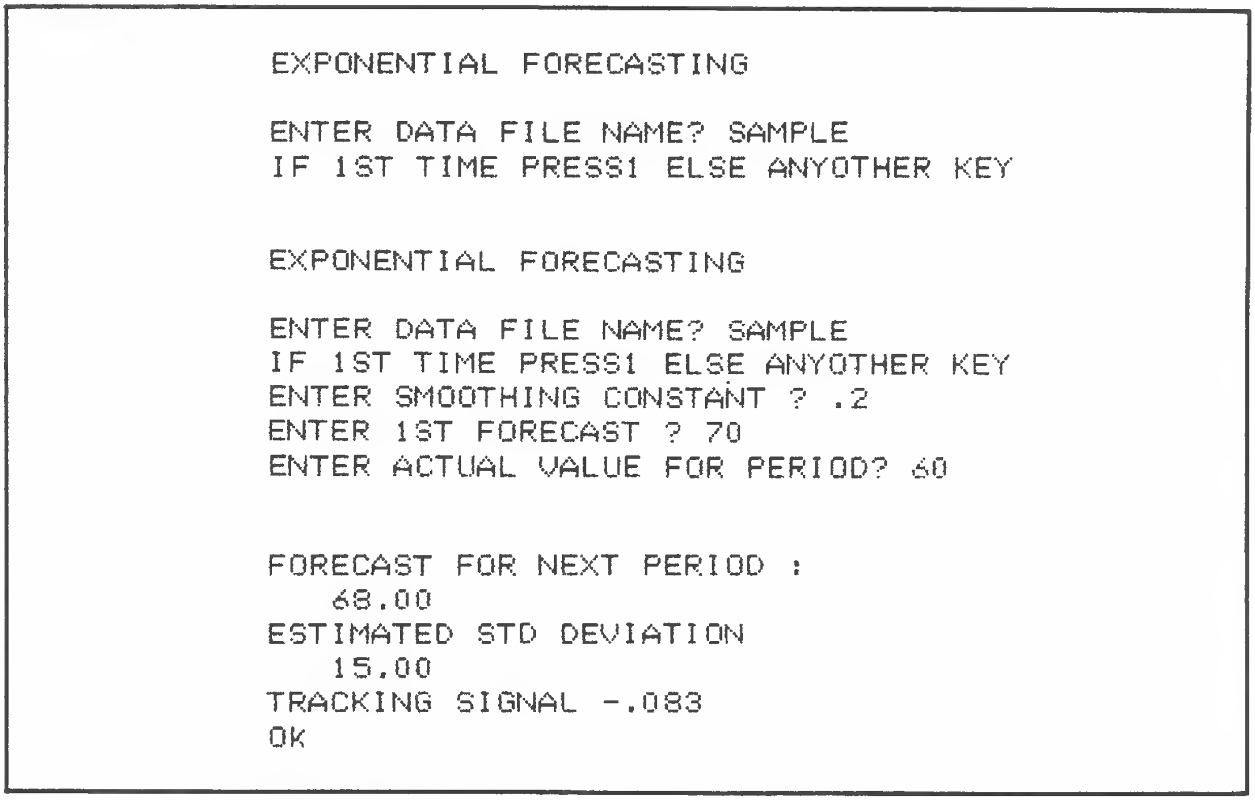
Once an out-of-control situation has been identified, it is essential to take corrective action. The first step is to determine why the forecast system is out of control. This is where special knowledge of your situation becomes important. As soon as the underlying reason for the situation is identified, additional corrective action can be taken.

You must remember that, in addition to correcting the forecasts to regain control, you must review and revise as necessary all the actions you took based on the forecasts. This step should not be neglected, but often is.

If the change in the data being forecast is judged to be temporary, you should be very careful about making any changes in the forecast. However, if you feel that the change is permanent, you should revise the entire forecast system to regain control. You then continue to forecast with the new system until another out of control situation is detected.

Example of ADJFOR.BA

The use of the adjustable exponential smoothing program can best be demonstrated with an example. Suppose that you are forecasting sales for the XY Widget Company. The first month's sales were 60 units. We will estimate that the forecast for the first month would have been 70 units. Now we have all the information needed to start the forecast



```
EXPONENTIAL FORECASTING

ENTER DATA FILE NAME? SAMPLE
IF 1ST TIME PRESS1 ELSE ANYOTHER KEY


EXPONENTIAL FORECASTING

ENTER DATA FILE NAME? SAMPLE
IF 1ST TIME PRESS1 ELSE ANYOTHER KEY
ENTER SMOOTHING CONSTANT ? .2
ENTER 1ST FORECAST ? 70
ENTER ACTUAL VALUE FOR PERIOD? 60


FORECAST FOR NEXT PERIOD :
   68.00
ESTIMATED STD DEVIATION
   15.00
TRACKING SIGNAL -.083
OK
```

Fig. 7-6. Data entry to initialize program ADJFOR.BA.

system. We will use a smoothing factor of 0.2 and the data will be stored in a RAM file called SALFOR. The complete data entry for the initialization of the program is shown in the Fig. 7-6. Note that the forecast for February is 68.

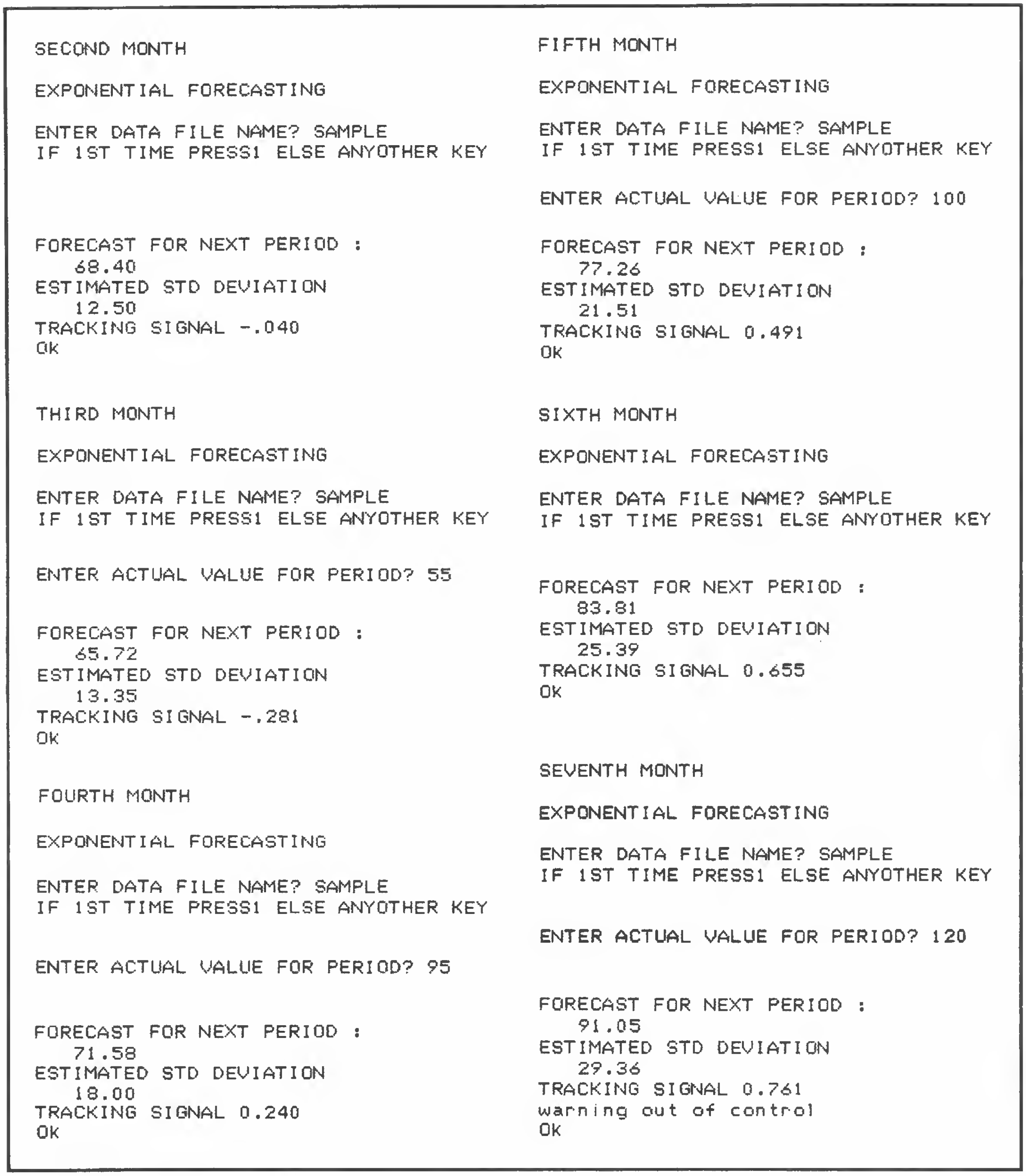
At the end of February we collect the sales figures and discover that the actual sales were 70. We now want to forecast the March demand. The only data we need to enter are the actual sales for February which was 70 units. The computer then calculates the March sales and prints 68.4 with a tracking signal of—0.04—which indicates that the system is in control. The March figures come in and show that sales were 55 units. We now forecast April and get 65.7 with a tracking signal of—0.281, which indicates that the system is still in control. Actual sales for April are 95 units. We then forecast May and get 71.58 and a tracking signal of 0.240. The system is still in control.

As soon as the May sales figures come in, we are ready to forecast June. May sales were 100 units, which we enter to forecast June. We get 77.26 with a tracking signal of 0.491, so the system is still in control. June sales were 110 units, which we enter to forecast July. The forecast for July is 83.81 with a tracking signal of 0.655. This is almost out of control, perhaps action should be taken. However, we will continue for one more period.

July sales are 120 units. We use this to forecast August sales and get predicted sales of 91 units and a tracking signal of 0.761. A warning message is printed, informing us that the system is out of control.

Before we forecast August sales, we need to take corrective action. First we determine why sales are higher than expected. We find that a new industry has moved into the area; the presence of additional people employed there provides a logical reason for the higher-than-expected sales. We thus assume that the increase is permanent.

We make a note to revise our long-range plans to account for the higher sales. Now we are ready to



```
SECOND MONTH

EXPONENTIAL FORECASTING

ENTER DATA FILE NAME? SAMPLE
IF 1ST TIME PRESS1 ELSE ANYOTHER KEY



FORECAST FOR NEXT PERIOD :
   68.40
ESTIMATED STD DEVIATION
   12.50
TRACKING SIGNAL -.040
Ok


THIRD MONTH

EXPONENTIAL FORECASTING

ENTER DATA FILE NAME? SAMPLE
IF 1ST TIME PRESS1 ELSE ANYOTHER KEY

ENTER ACTUAL VALUE FOR PERIOD? 55


FORECAST FOR NEXT PERIOD :
   65.72
ESTIMATED STD DEVIATION
   13.35
TRACKING SIGNAL -.281
Ok


FOURTH MONTH

EXPONENTIAL FORECASTING

ENTER DATA FILE NAME? SAMPLE
IF 1ST TIME PRESS1 ELSE ANYOTHER KEY

ENTER ACTUAL VALUE FOR PERIOD? 95


FORECAST FOR NEXT PERIOD :
   71.58
ESTIMATED STD DEVIATION
   18.00
TRACKING SIGNAL 0.240
Ok
```

```
FIFTH MONTH

EXPONENTIAL FORECASTING

ENTER DATA FILE NAME? SAMPLE
IF 1ST TIME PRESS1 ELSE ANYOTHER KEY

ENTER ACTUAL VALUE FOR PERIOD? 100

FORECAST FOR NEXT PERIOD :
   77.26
ESTIMATED STD DEVIATION
   21.51
TRACKING SIGNAL 0.491
Ok


SIXTH MONTH

EXPONENTIAL FORECASTING

ENTER DATA FILE NAME? SAMPLE
IF 1ST TIME PRESS1 ELSE ANYOTHER KEY


FORECAST FOR NEXT PERIOD :
   83.81
ESTIMATED STD DEVIATION
   25.39
TRACKING SIGNAL 0.655
Ok


SEVENTH MONTH

EXPONENTIAL FORECASTING

ENTER DATA FILE NAME? SAMPLE
IF 1ST TIME PRESS1 ELSE ANYOTHER KEY

ENTER ACTUAL VALUE FOR PERIOD? 120


FORECAST FOR NEXT PERIOD :
   91.05
ESTIMATED STD DEVIATION
   29.36
TRACKING SIGNAL 0.761
warning out of control
Ok
```

Fig. 7-7. Month-by-month prediction.

take action to correct the forecast. The first step is to identify when the change actually took place. As we look at the data we see that sales jumped from 55 in March to 95 in April and continued at a high level from April on. Thus we assume that the change took place in April. We begin our new forecasting system by manually adjusting the April forecast, as shown in Fig. 7-8. Since this is a new forecasting system, we press 1 to tell the computer that this is the first forecast for the system. We will assume a forecast level of sales for April of 100 units to start the system. (Note that we get a revised forecast for May of 99 units and a zero tracking signal.)

We now use the actual May data to prepare the revised forecast for June, and then the actual June data to get the revised forecast for July—all of which are shown in the printout. We are now ready to forecast August. We get a forecast of 105 units and a tracking signal of 0.498. Sales for August are actually 100 units, which we use to forecast sales for September. As soon as we have the figures for September sales, we use the program to forecast October, and so on.

Note that the system was actually out of control for April, May, June, and July, and that we detected the out-of-control situation in July. In practice we would have detected the out-of-control situation in June, because we should round the figures for the control signal to one significant figure. However, I find that for debugging published programs it is important to have more significant figures than you would use when you actually run the program. Small errors keying in equations often will not show up under normal debugging if only one significant figure is used.

You can also put tighter control on your forecast by using a smaller value of the tracking signal to indicate an out-of-control situation. If the critical tracking signal is 0.5, there is a 90 percent probability that the system is out of control when the smoothing constant is 0.2, and about an 80 percent probability that the system is out of control if the smoothing constant is 0.3. The critical value of the tracking signal for 95 percent probability of the system being out of control is 0.42 for A=1, 0.58 for A=0.2, 0.71 for A=0.3, and 0.82 for A=0.4.

Notes on the Program

The adaptive exponential smoothing program is well suited for use on the Model 100. The program requires a minimum of data and thus the data can reside in RAM files and never be noticed by other programs. The program produces good short-term forecasts and provides a warning when the system is out of control.

One of the better uses for the program is to forecast demand for all the items in inventory. Each item will have a forecast and a tracking signal. The tracking signal can be used to identify situations where management attention is needed. Management thus can concentrate on those few items that really require attention.

As mentioned above, when you use the program, you should round the tracking signal to one significant figure. However, when you debug the program you should use all the significant figures in the example.

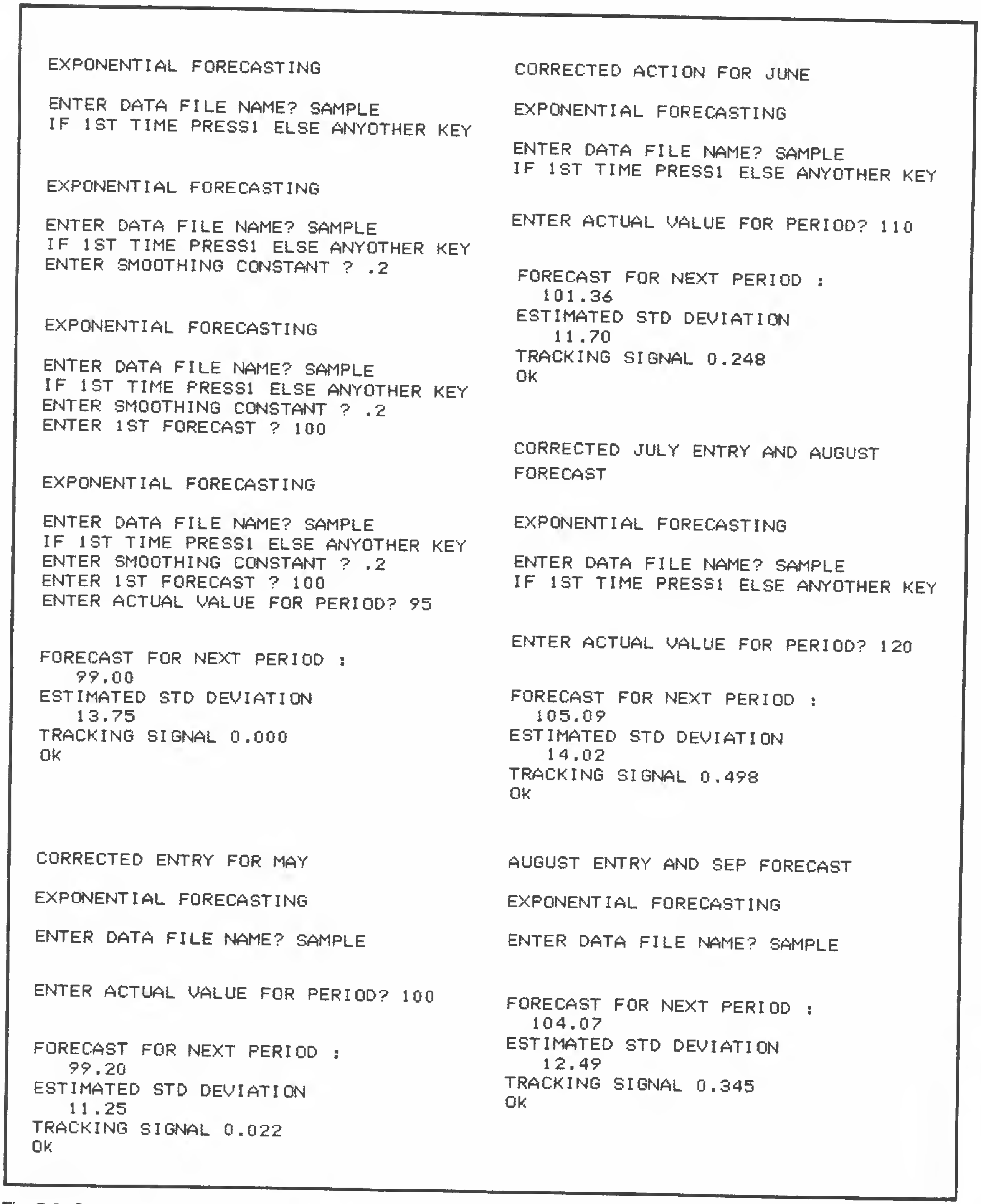
Moving Average Smoothing

Moving average smoothing is another popular method of smoothing time series. The moving average is calculated by replacing the data by the average of a given fixed number of observations. The forecast for the next period is the average of the last set of observations. The number of observations included in the average is set by the user and is held constant during the analysis. The more observations included in the average, the smoother the analysis will be.

Moving average analysis is used in many systems of technical analysis of the stock market. One of the simplest techniques is to buy stocks when the price moves above the moving average curve and sell when the price moves below it.

One of the big drawbacks of the moving average technique is the large amount of data that must be stored to use the method. For example, if you use the 50-day moving average of stock prices, you must store the last 50 days of data. Many moving averages techniques use 100-day or 200-day moving averages, which means that 100 to 200 data points must be stored. Such large storage requirements are not a serious problem with microcomputers with disk drives, but the data requirements can be a problem for those of us using the Model 100.

Because exponential smoothing gives the



```
EXPONENTIAL FORECASTING

ENTER DATA FILE NAME? SAMPLE
IF 1ST TIME PRESS1 ELSE ANYOTHER KEY


EXPONENTIAL FORECASTING

ENTER DATA FILE NAME? SAMPLE
IF 1ST TIME PRESS1 ELSE ANYOTHER KEY
ENTER SMOOTHING CONSTANT ? .2


EXPONENTIAL FORECASTING

ENTER DATA FILE NAME? SAMPLE
IF 1ST TIME PRESS1 ELSE ANYOTHER KEY
ENTER SMOOTHING CONSTANT ? .2
ENTER 1ST FORECAST ? 100


EXPONENTIAL FORECASTING

ENTER DATA FILE NAME? SAMPLE
IF 1ST TIME PRESS1 ELSE ANYOTHER KEY
ENTER SMOOTHING CONSTANT ? .2
ENTER 1ST FORECAST ? 100
ENTER ACTUAL VALUE FOR PERIOD? 95


FORECAST FOR NEXT PERIOD :
   99.00
ESTIMATED STD DEVIATION
   13.75
TRACKING SIGNAL 0.000
Ok



CORRECTED ENTRY FOR MAY

EXPONENTIAL FORECASTING

ENTER DATA FILE NAME? SAMPLE


ENTER ACTUAL VALUE FOR PERIOD? 100


FORECAST FOR NEXT PERIOD :
   99.20
ESTIMATED STD DEVIATION
   11.25
TRACKING SIGNAL 0.022
Ok
```

```
CORRECTED ACTION FOR JUNE

EXPONENTIAL FORECASTING

ENTER DATA FILE NAME? SAMPLE
IF 1ST TIME PRESS1 ELSE ANYOTHER KEY


ENTER ACTUAL VALUE FOR PERIOD? 110


FORECAST FOR NEXT PERIOD :
   101.36
ESTIMATED STD DEVIATION
    11.70
TRACKING SIGNAL 0.248
Ok


CORRECTED JULY ENTRY AND AUGUST
FORECAST

EXPONENTIAL FORECASTING

ENTER DATA FILE NAME? SAMPLE
IF 1ST TIME PRESS1 ELSE ANYOTHER KEY


ENTER ACTUAL VALUE FOR PERIOD? 120


FORECAST FOR NEXT PERIOD :
   105.09
ESTIMATED STD DEVIATION
    14.02
TRACKING SIGNAL 0.498
Ok



AUGUST ENTRY AND SEP FORECAST

EXPONENTIAL FORECASTING

ENTER DATA FILE NAME? SAMPLE


FORECAST FOR NEXT PERIOD :
   104.07
ESTIMATED STD DEVIATION
    12.49
TRACKING SIGNAL 0.345
Ok
```

Fig. 7-8. Correcting an out-of-control prediction sequence.

same type of smoothing as does moving average, and because exponential smoothing requires much less data, it is generally preferred over simple moving average smoothing.

ADAPTIVE FILTERING

Another objection to simple moving average smoothing is that it gives all the past data points the same weight. For example a forecast based on a 50-day moving average assigns as much importance to the data 50 days ago as it does to the data from yesterday. Intuitively, it seems that the data from yesterday should be more important for today than the data from 50 days ago.

Forecasting techniques have been developed to weight the data to give more importance to some data than others. Such techniques are called *weighted moving average smoothing.* Generally, if you can determine the proper weights to assign to the data, you can produce very good forecasts using weighted moving averages. However, there is the problem of picking the proper weights.

The program ADAPT.BA (Listing 7-5) is designed to do the hard work for you. The program can produce forecasts with an accuracy that rivals that of any other forecasting technique.

The disadvantages of the method are:

1) Large amounts of data are required to use it.
2) Considerable time is required to do the calculations.

Generally these two disadvantages would rule out the technique for use on the Model 100. However, the accuracy of the forecasts is enough to overcome the disadvantages. The accuracy will be demonstrated in the examples.

Two examples are presented showing how to use ADAPT.BA. The first example is a debugging example. It runs fairly fast and will enable you to debug the program without waiting long hours for the program to finish its calculations. The second example is intended to show off the program's forecasting ability. The program uses quarterly earnings data from a real company and attempts to forecast the earnings for the next four quarters. As you can see from the example, the largest error in the forecast was less than 8 percent. This example requires about 4.5 hours to run.

Using ADAPT.BA

The program ADAPT.BA uses the adaptive filtering technique to analyze time series data and to prepare forecasts based on the analysis. The program reads data from a data file created by the ENTER.BA program. The past data are used to estimate the weights that should be used to make the forecast, based on weighted moving average analysis. The weighted moving average is calculated with a user-specified number of observations (periods) that are to be used in calculating the moving average. The number of observations to be included in the moving average should be selected to ensure that both seasonal and cyclic features of the time series are included.

For example, assume that you are forecasting quarterly sales. The past data show that your business has two seasons per year. Additional analysis shows that the data also have a four-year cycle. You would use a 16-quarter moving average for your analysis. If there were no cycle in the data, you could use a 4 quarter average.

The data required by the program are:

1) The time series data, which must include more data than the number of periods selected for averaging. In our example above, we would like to have at least 20 observations for the 4-year cyclic case. (Using 32 observations to cover two complete cycles would be still better.) We should have at least 8 observations for the no-cycle case, but, again, more data would be better.
2) The user-specified number of periods to be used in the averaging process.
3) The name of the file where the data are stored.

When the program has finished calculating the optimum weights, it will ask you to specify the number of periods to be forecast. The program will then calculate and print on the display the forecast for each period. The forecasts should be updated as new data are added to the data base. Two examples showing the use of the program follow Example 1 is intended to help you debug the program.

Example 1

Enter the data shown in Table 7-1, using the program ENTER.BA. The data entry process is illustrated in Fig. 7-9. Note that the data are given the file name TEST, the X variable is called X, and the Y variable is called Y. When all the data have been entered, enter−99 for X, press ENTER and then press ∗ to store the data.

As soon as the data are stored, run ADAPT.BA. The entire run is shown in Figs. 7-10 and 7-11. Note that we use two periods for the training. While the computer is calculating, it will display the note WORKING and it will tell you which iteration is working on. The program is set up to use a maximum of 200 iterations. This example requires about 10 minutes to run.

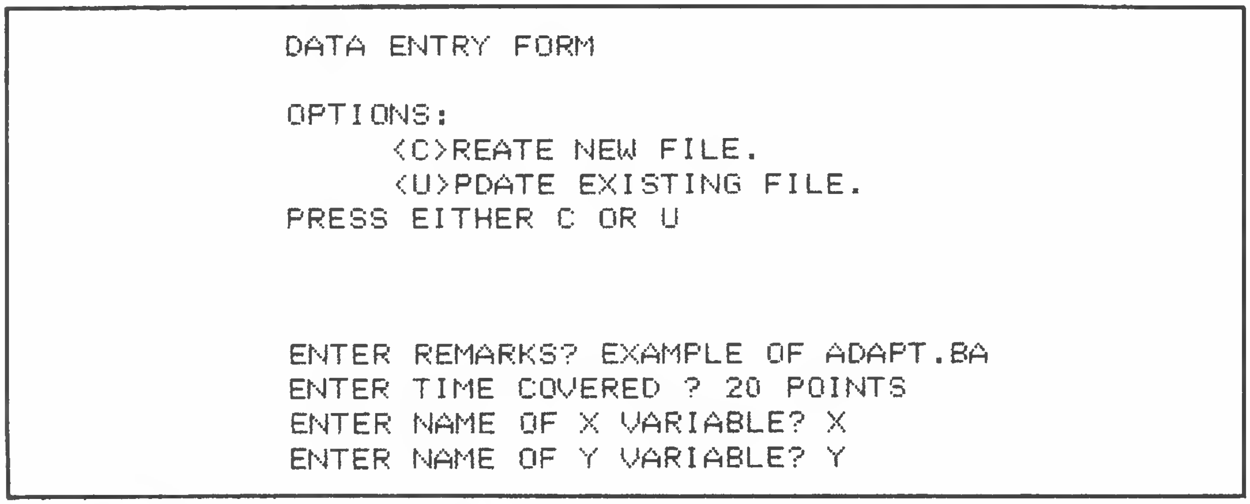
Example 2

This example demonstrates the accuracy of the forecasts produced by ADAPT.BA. The historical data are quarterly earnings and are given in Table 7-2. These are actual earnings for a major corporation. The format for the year and quarter is *year.quarter*. For example, 1960.1 is the first quarter of 1960. I used a 20-period analysis. It appears that 20 periods (five years) should cover the cycle of the data. I asked for a four-period (one year) forecast. The entire example is shown in Fig. 7-12.

Table 7-1. Sample Data for Example 1.

X	Y
1	2
2	4
3	6
4	8
5	10
6	12
7	14
8	16
9	18
10	20
11	22
12	24
13	26
14	28
15	30
16	32
17	34
18	36
19	38
20	40

As you can see from the figure, the forecast earnings are in very good agreement with the actual earnings. The maximum error is in the fourth



```
DATA ENTRY FORM

OPTIONS:
     <C>REATE NEW FILE.
     <U>PDATE EXISTING FILE.
PRESS EITHER C OR U



ENTER REMARKS? EXAMPLE OF ADAPT.BA
ENTER TIME COVERED ? 20 POINTS
ENTER NAME OF X VARIABLE? X
ENTER NAME OF Y VARIABLE? Y
```

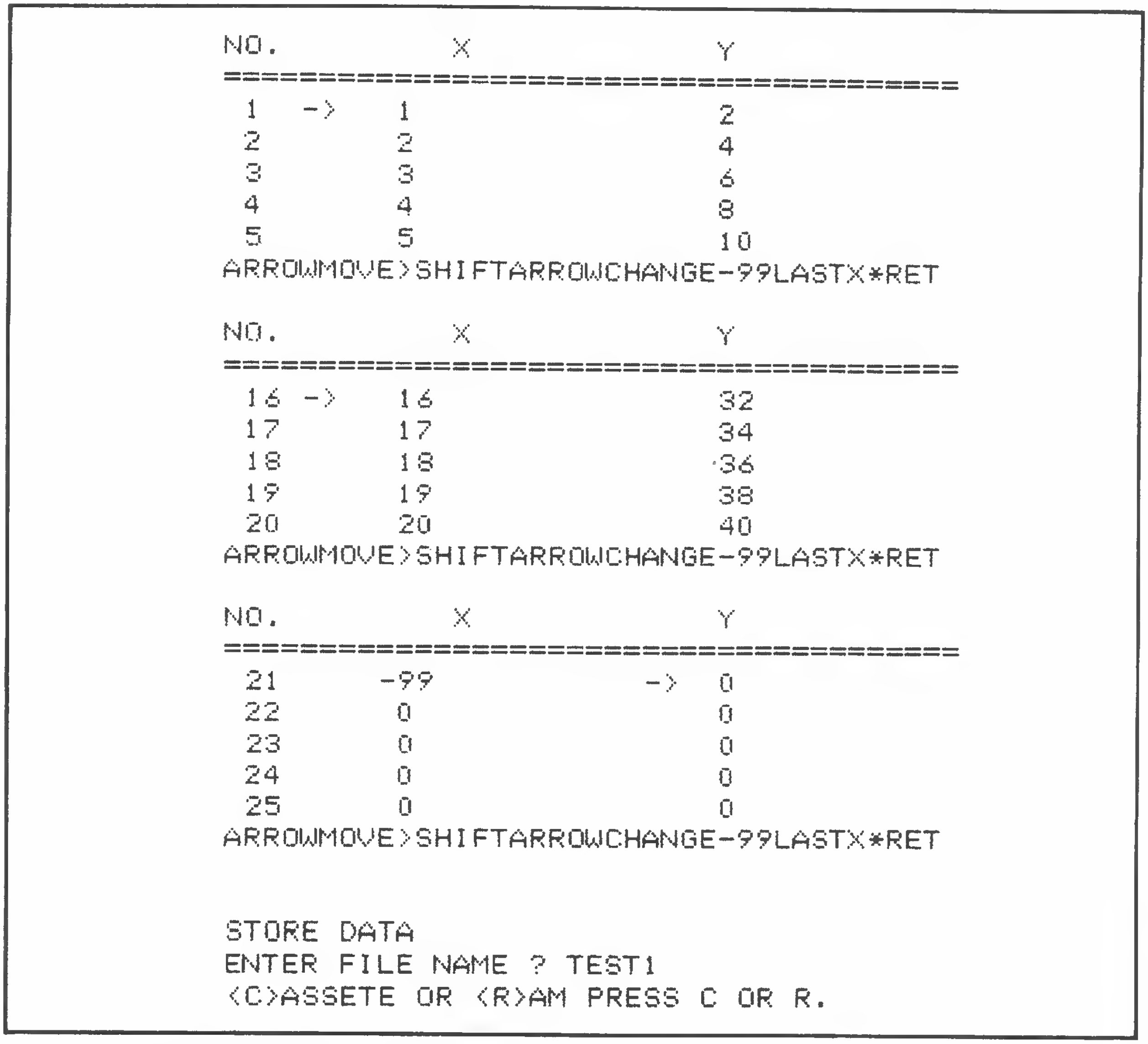
Fig. 7-9. Data entry for adaptive filtering example.

quarter, but this error is still less than 8 percent. This example required about 4.5 hours to run.

General Comments

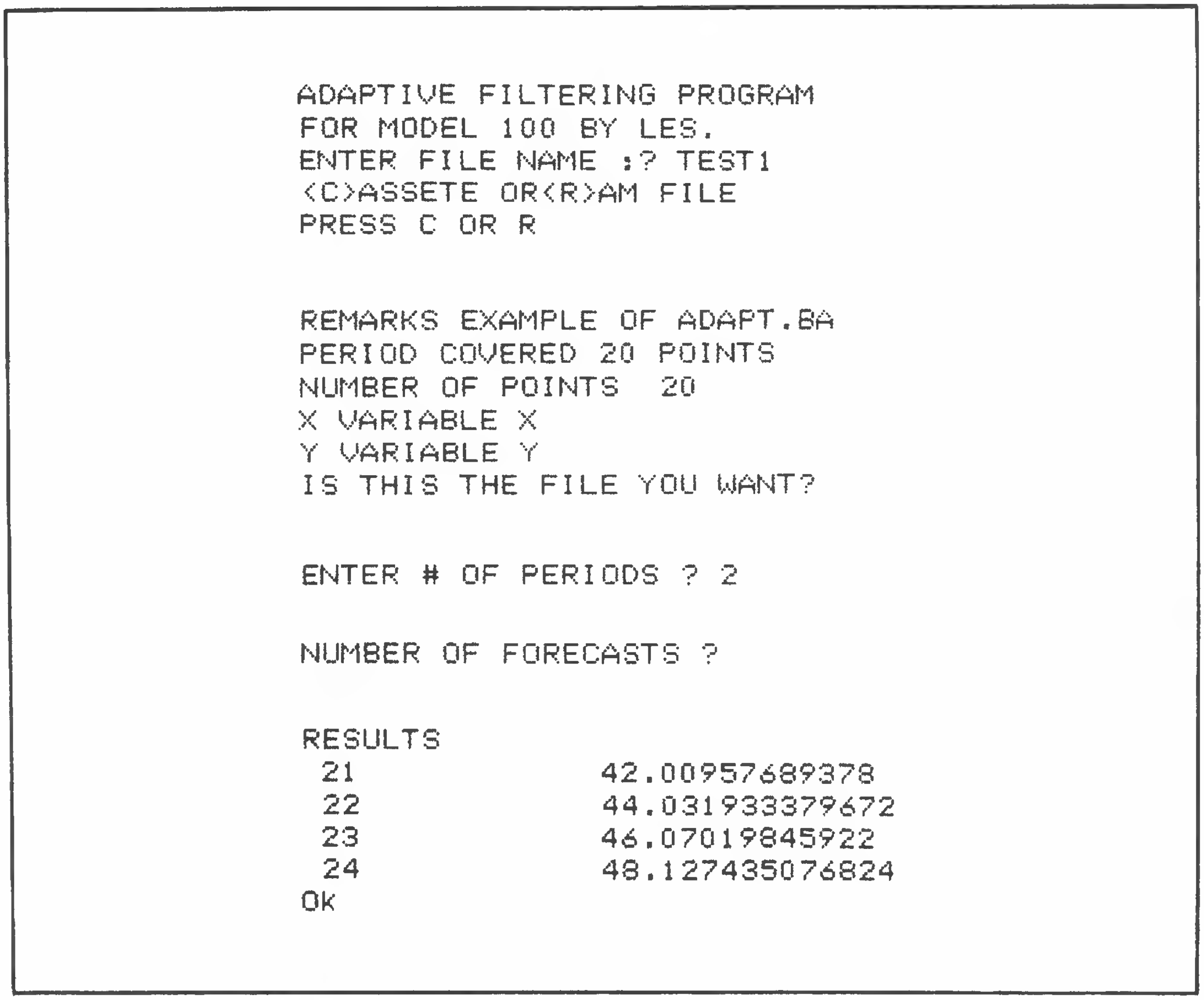
Proper selection of the number of periods to use in the analysis is an important part of using the adaptive filtering program. The best way to determine how many periods to include is to plot the data and look for seasonal and cyclic patterns. A plot of the data for Example 2 of the ADAPT.BA is shown in Fig. 7-13. Note that there is a definite seasonal pattern. There is also a trend in increasing earnings. Finally, there appears to be a cycle of four to five years in the data. Since we want to be sure that the entire cycle is covered in the analysis, five years or 20 quarters appears to be a reasonable value for the number of periods to include in the analysis.

You can plot the data either on graph paper or on your Model 100 display, using the plot program included in this chapter. The plotting program should be used as the first step in analyzing data,



```
NO.           X             Y
========================================
 1   ->   1                 2
 2        2                 4
 3        3                 6
 4        4                 8
 5        5                 10
ARROWMOVE>SHIFTARROWCHANGE-99LASTX*RET

NO.           X             Y
========================================
 16  ->   16                32
 17       17                34
 18       18                36
 19       19                38
 20       20                40
ARROWMOVE>SHIFTARROWCHANGE-99LASTX*RET

NO.           X             Y
========================================
 21       -99           ->  0
 22       0                 0
 23       0                 0
 24       0                 0
 25       0                 0
ARROWMOVE>SHIFTARROWCHANGE-99LASTX*RET


STORE DATA
ENTER FILE NAME ? TEST1
<C>ASSETE OR <R>AM PRESS C OR R.
```

Fig. 7-10. Beginning of example run of ADAPT.BA.



```
ADAPTIVE FILTERING PROGRAM
FOR MODEL 100 BY LES.
ENTER FILE NAME :? TEST1
<C>ASSETE OR<R>AM FILE
PRESS C OR R


REMARKS EXAMPLE OF ADAPT.BA
PERIOD COVERED 20 POINTS
NUMBER OF POINTS  20
X VARIABLE X
Y VARIABLE Y
IS THIS THE FILE YOU WANT?


ENTER # OF PERIODS ? 2


NUMBER OF FORECASTS ?


RESULTS
 21              42.00957689378
 22              44.031933379672
 23              46.07019845922
 24              48.127435076824
Ok
```

Fig. 7-11. Completion of example run for ADAPT.BA.

regardless of the type of analysis. The plotting program can help you determine if regression analysis is appropriate for the data, in addition to helping you determine the type of time series analysis to use.

Differences

Most forecasting techniques work best when there is no trend in the data. Therefore, it is often beneficial to remove the trend before attempting to make a forecast. In fact, many commercial forecasting packages provide programs to calculate the differences.

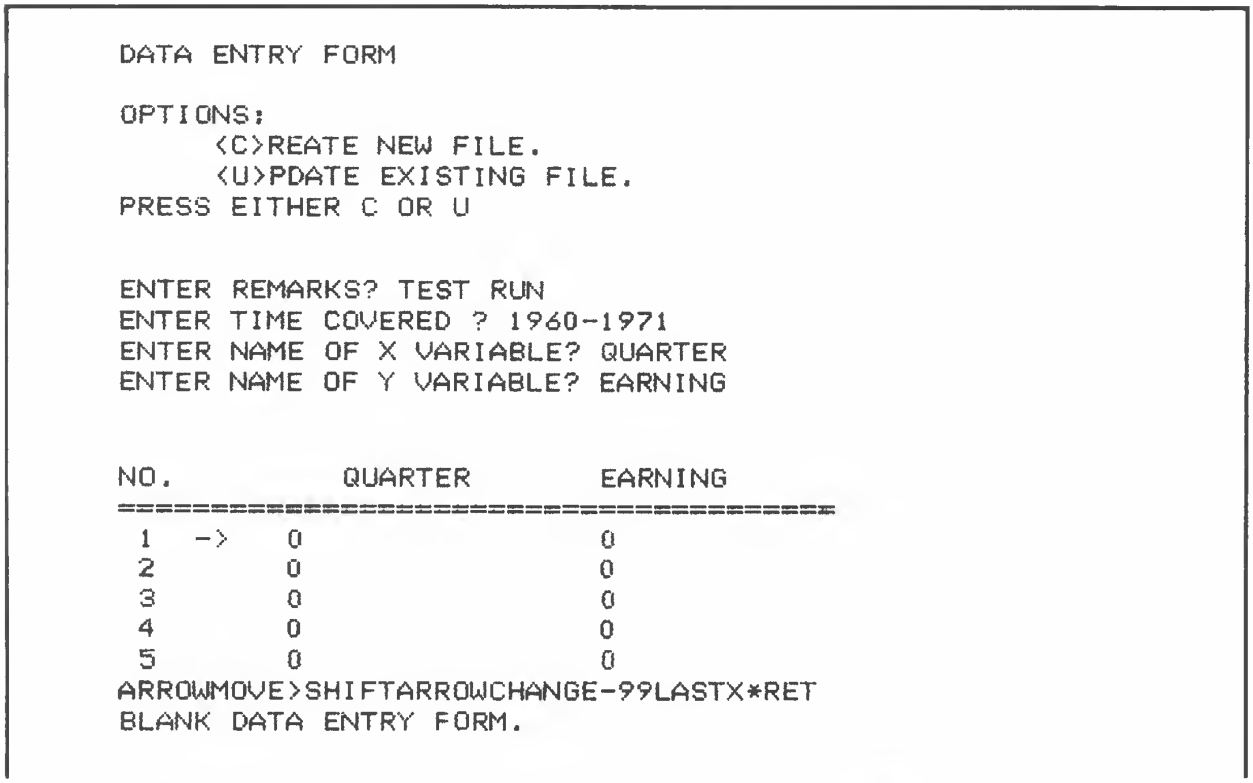
The differences are calculated by subtracting the value of the forecast variable at time t-1 from the value at time t. For example, if last month's sales were 100 and this month's sales were 110, the difference would be calculated as

$$D = 110 - 100 = 10$$

We can forecast the difference between next month's sales and this month's sales. The forecast for the actual value of sales is then this month's sales plus the forecast difference. You can write a short program to calculate the differences and then use the differences in the earnings forecast example. You'll find an improvement in the forecast.

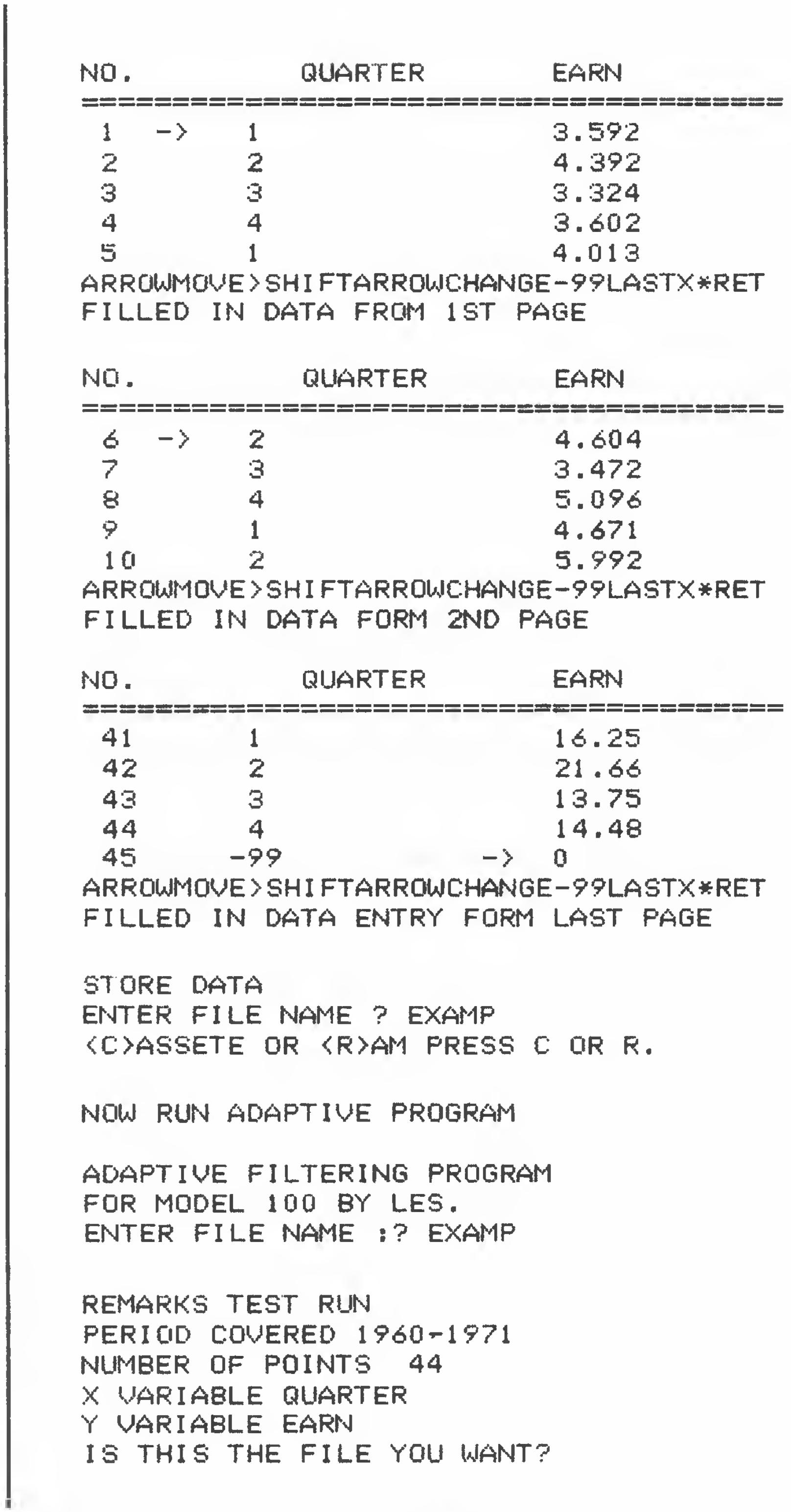
Table 7-2. Adaptive Filtering Using Actual Data for a Major Corporation.

YEAR.QUARTER	EARNINGS$1,000,000	YEAR.QUARTER	EARNINGS$1.000.000
1960.1	3.592	1965.3	6.14
1960.2	4.392	1965.4	7.085
1960.3	3.324	1966.1	9.49
1960.4	3.602	1966.2	11.6
1961.1	4.013	1966.3	7.579
1961.2	4.604	1966.4	9.465
1961.3	3.472	1967.1	10.2
1961.4	5.096	1967.2	12.45
1962.1	4.671	1967.3	8.159
1962.2	5.992	1967.4	9.799
1962.3	4.249	1968.1	11.47
1962.4	5.472	1968.2	13.5
1963.1	5.994	1968.3	8.726
1963.2	7.778	1968.4	10.16
1963.3	5.906	1969.1	13.35
1963.4	8.047	1969.2	17.07
1964.1	6.592	1969.3	11.72
1964.2	8.365	1969.4	14.48
1964.3	6.483	1970.1	16.25
1964.4	7.016	1970.2	21.66
1965.1	7.879	1970.3	13.75
1965.2	11.13	1970.4	14.48



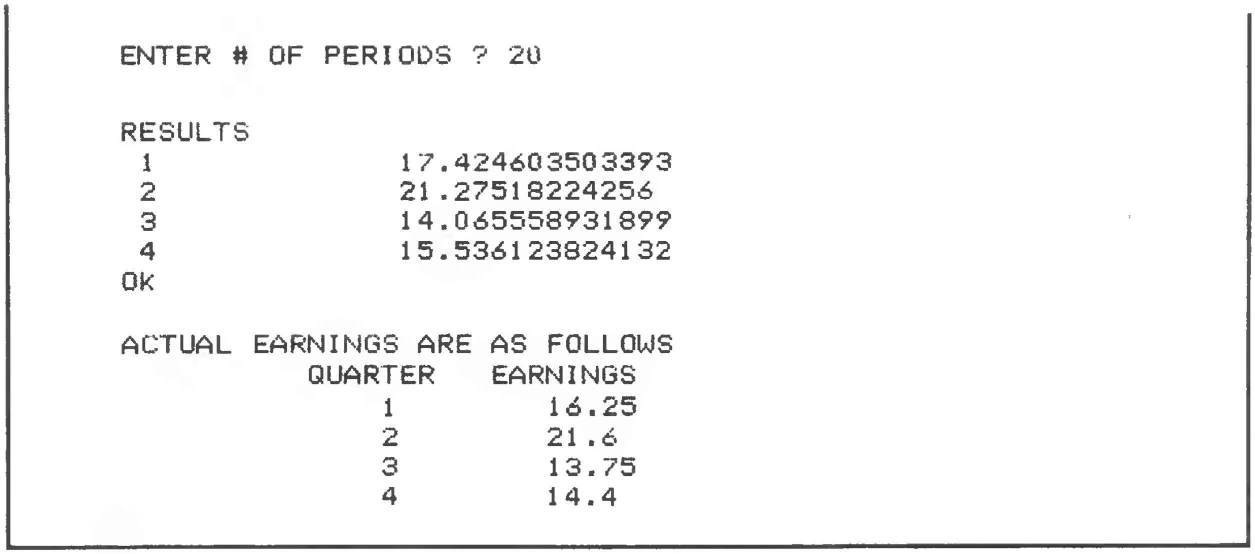
```
DATA ENTRY FORM

OPTIONS:
      <C>REATE NEW FILE.
      <U>PDATE EXISTING FILE.
PRESS EITHER C OR U


ENTER REMARKS? TEST RUN
ENTER TIME COVERED ? 1960-1971
ENTER NAME OF X VARIABLE? QUARTER
ENTER NAME OF Y VARIABLE? EARNING


NO.           QUARTER        EARNING
=========================================
 1   ->    0                   0
 2         0                   0
 3         0                   0
 4         0                   0
 5         0                   0
ARROWMOVE>SHIFTARROWCHANGE-99LASTX*RET
BLANK DATA ENTRY FORM.
```

Fig. 7-12. Adaptive weighted moving average forecasting with real data.(cont'd on page 118).

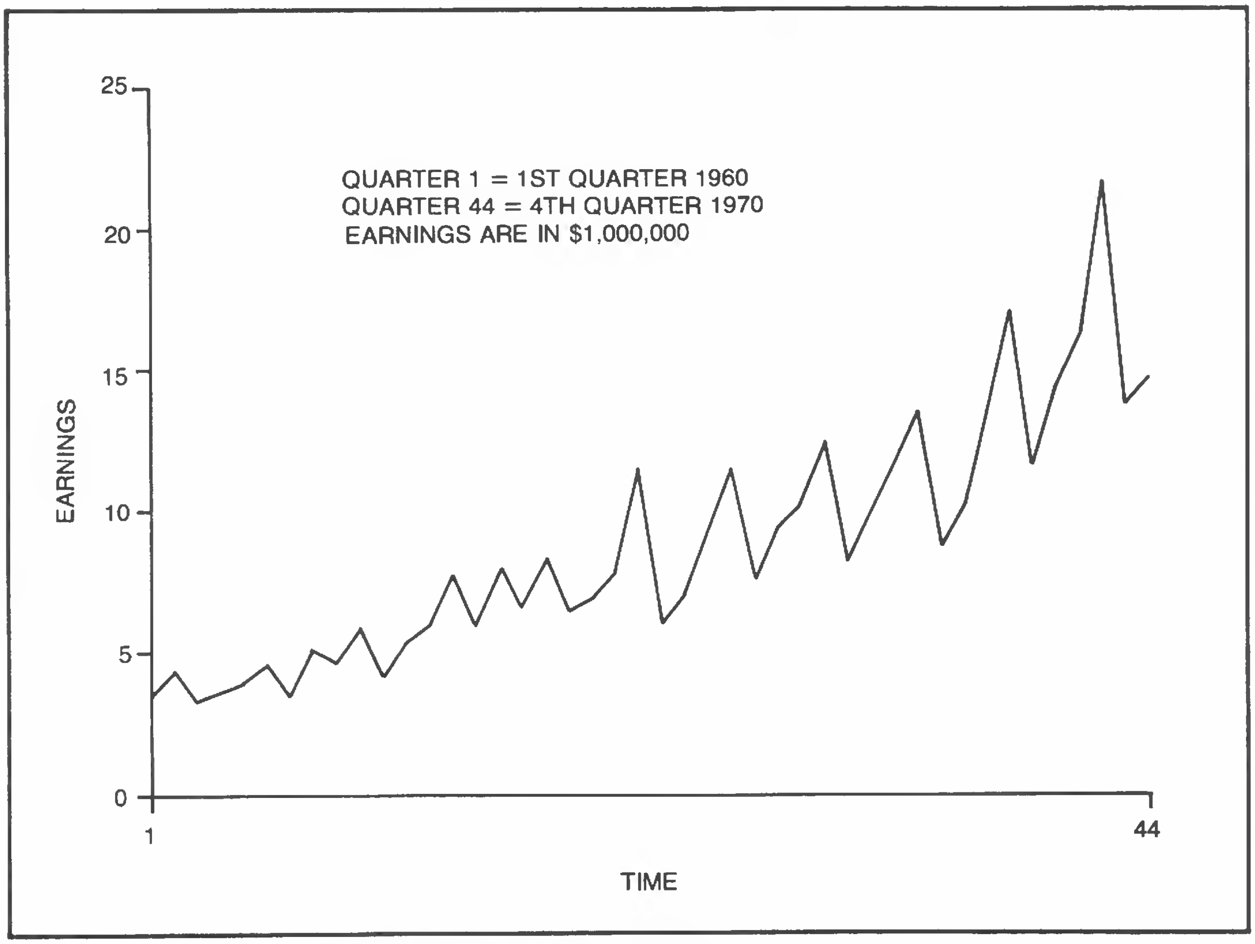


```
NO.           QUARTER        EARN
=======================================
 1   ->   1                  3.592
 2        2                  4.392
 3        3                  3.324
 4        4                  3.602
 5        1                  4.013
ARROWMOVE>SHIFTARROWCHANGE-99LASTX*RET
FILLED IN DATA FROM 1ST PAGE

NO.           QUARTER        EARN
=======================================
 6   ->   2                  4.604
 7        3                  3.472
 8        4                  5.096
 9        1                  4.671
 10       2                  5.992
ARROWMOVE>SHIFTARROWCHANGE-99LASTX*RET
FILLED IN DATA FORM 2ND PAGE

NO.           QUARTER        EARN
=======================================
 41       1                  16.25
 42       2                  21.66
 43       3                  13.75
 44       4                  14.48
 45      -99             ->  0
ARROWMOVE>SHIFTARROWCHANGE-99LASTX*RET
FILLED IN DATA ENTRY FORM LAST PAGE

STORE DATA
ENTER FILE NAME ? EXAMP
<C>ASSETE OR <R>AM PRESS C OR R.

NOW RUN ADAPTIVE PROGRAM

ADAPTIVE FILTERING PROGRAM
FOR MODEL 100 BY LES.
ENTER FILE NAME :? EXAMP

REMARKS TEST RUN
PERIOD COVERED 1960-1971
NUMBER OF POINTS  44
X VARIABLE QUARTER
Y VARIABLE EARN
IS THIS THE FILE YOU WANT?
```

Fig. 7-12. Adaptive weighted moving average forecasting with real data (cont'd on page 119).



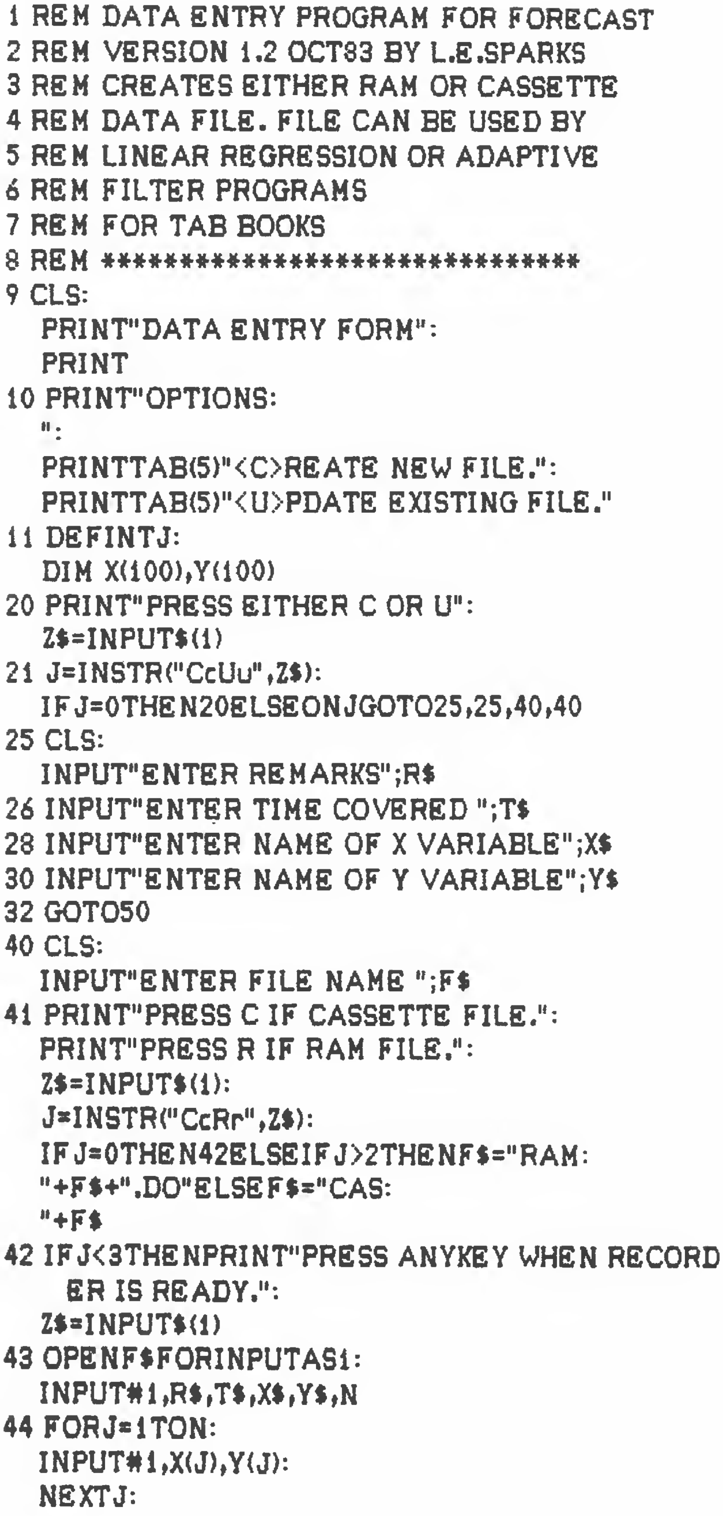
```
ENTER # OF PERIODS ? 20

RESULTS
 1              17.424603503393
 2              21.27518224256
 3              14.065558931899
 4              15.536123824132
OK

ACTUAL EARNINGS ARE AS FOLLOWS
          QUARTER    EARNINGS
             1          16.25
             2          21.6
             3          13.75
             4          14.4
```



QUARTER 1 = 1ST QUARTER 1960
QUARTER 44 = 4TH QUARTER 1970
EARNINGS ARE IN $1,000,000

Fig. 7-13. Plot of predicted quarterly earnings for a major corporation.

Listing 7-1. Data entry program for forecasting.



```
1 REM DATA ENTRY PROGRAM FOR FORECAST
2 REM VERSION 1.2 OCT83 BY L.E.SPARKS
3 REM CREATES EITHER RAM OR CASSETTE
4 REM DATA FILE. FILE CAN BE USED BY
5 REM LINEAR REGRESSION OR ADAPTIVE
6 REM FILTER PROGRAMS
7 REM FOR TAB BOOKS
8 REM ********************************
9 CLS:
   PRINT"DATA ENTRY FORM":
   PRINT
10 PRINT"OPTIONS:
   ":
   PRINTTAB(5)"<C>REATE NEW FILE.":
   PRINTTAB(5)"<U>PDATE EXISTING FILE."
11 DEFINTJ:
   DIM X(100),Y(100)
20 PRINT"PRESS EITHER C OR U":
   Z$=INPUT$(1)
21 J=INSTR("CcUu",Z$):
   IFJ=0THEN20ELSEONJGOTO25,25,40,40
25 CLS:
   INPUT"ENTER REMARKS";R$
26 INPUT"ENTER TIME COVERED ";T$
28 INPUT"ENTER NAME OF X VARIABLE";X$
30 INPUT"ENTER NAME OF Y VARIABLE";Y$
32 GOTO50
40 CLS:
   INPUT"ENTER FILE NAME ";F$
41 PRINT"PRESS C IF CASSETTE FILE.":
   PRINT"PRESS R IF RAM FILE.":
   Z$=INPUT$(1):
   J=INSTR("CcRr",Z$):
   IFJ=0THEN42ELSEIFJ>2THENF$="RAM:
   "+F$+".DO"ELSEF$="CAS:
   "+F$
42 IFJ<3THENPRINT"PRESS ANYKEY WHEN RECORD
     ER IS READY.":
   Z$=INPUT$(1)
43 OPENF$FORINPUTAS1:
   INPUT#1,R$,T$,X$,Y$,N
44 FORJ=1TON:
   INPUT#1,X(J),Y(J):
   NEXTJ:
    CLOSE
45 CLS:
    PRINTR$:
    PRINTT$:
    PRINT"NO POINTS :
    ";N:
    PRINT"PRESS Y TO CONTINUE ANYOTHER KEY
      QUITS.":
    Z$=INPUT$(1):
    IFZ$="Y"ORZ$="Y"THEN50ELSEEND
50 REM
51 A$="1234567890.+-"
52 AZ$=CHR$(13)+CHR$(31)+CHR$(30)+CHR$(28)
      +CHR$(29)+CHR$(42)+CHR$(2)+CHR$(20)
53 J1=1:
    J2 = 5
54 JS=2:
    JR=4
60 GOSUB9000
70 CLS:
    PRINT:
    PRINT"STORE DATA"
80 INPUT"ENTER FILE NAME ";F$
90 PRINT"<C>ASSETE OR <R>AM PRESS C OR R."

100 Z$=INPUT$(1):
    J1=INSTR("CcRr",Z$):
    IFJ1=0THEN90
110 IFJ1>2THENF1$="RAM:
    "ELSEF1$="CAS:
    "
120 IFJ1<3THENPRINT"PRESS ANYKEY WHEN RECO
      RDER IS READY.":
    Z$=INPUT$(1)ELSEF$=F$+".DO"
122 F1$=F1$+F$:
     OPENF1$FOROUTPUT AS1:
    PRINT#1,R$;",";T$;",";X$;",";Y$;",";N
130 FORJ=1TON:
    PRINT#1,X(J),Y(J):
    NEXTJ:
    CLOSE:
    END
9000 REM DATA INPUT
9010 CLS:
    PRINT "NO.";TAB(12)X$;TAB(26);Y$
```

```
9020 PRINT STRING$(39,"=")
9040 FOR J=J1TOJ2
9050    PRINT J;TAB(8);X(J);TAB(25);Y(J)
9060 NEXT J
9070 PRINT @(JS*40)+JR,"->";
9075 PRINT@280,"ARROWMOVE>SHIFTARROWCHANGE
      -99LASTX*RET";
9080 Y1$=INPUT$(1)
9090 IFINSTR(A$,Y1$)<>0 THEN 9500
9100 JZ=INSTR(AZ$,Y1$):
   IFJZ=0THEN9080
9110 ON JZ GOTO 9120,9120,9140,9160,9180,9
      199,9200,9250
9120 PRINT@(JS*40)+JR,"    ";
9122 JS=JS+1:
   IFJS>6THEN JS=2:
9124 PRINT@(JS*40)+JR,"->";
9130 GOTO 9080
9140 PRINT@(40*JS)+JR,"   ";
9150 JS =JS-1:
   IFJS<2 THEN JS=6
9152 PRINT@(40*JS)+JR,"->";:
   GOTO9080
9160 PRINT@(JS*40)+JR,"    ";
9170 IF JR=4 THEN JR=22 ELSEJR=4
9175 PRINT@(JS*40)+JR,"->";:
   GOTO 9080
9180 PRINT@(JS*40)+JR,"  ";
9185 IF JR=22 THEN JR=4 ELSE JR=22
9190 PRINT@(JS*40)+JR,"->";:
   GOTO 9080
9199 RETURN
9200 J1 = J2+1:
   J2=J1+4
9210 JR = 4:
    JS=2
9220 GOTO 9010
9250 IF J1 = 1 THEN 9010
9260 J2 = J1-1:
   J1 = J2 - 4
9270 GOTO 9010
9500 PRINT @(40*JS)+JR,"   ";
9505 PRINT@280,"ENTER REST OF DATA PRESSEN
      TER WHEN DONE";
9510 V$=Y1$
```

```
9530 PRINT@(40*JS)+JR,V$;CHR$(95);"  ";
9540 Y1$=INKEY$:
   IF Y1$ =""THEN 9540
9550 IF Y1$=CHR$(13) THEN 9700
9560 IF Y1$<>CHR$(29) THEN 9600
9570 IF LEN (V$)=0 THEN 9530
9580 V$=LEFT$(V$,LEN(V$)-1):

9590 Y1$="":
   GOTO 9530
9600 IF INSTR(A$,Y1$)=0 THEN 9530
9610 V$ = V$ + Y1$
9620 GOTO 9530
9700 IF JR = 4THEN X(J1+JS-2)=VAL(V$):
   IFX(J1+JS-2)=-99THENN=J1+JS-3:

9701 IFJR=4THENJR=22:
   GOTO9010
9710 Y(J1+JS-2)=VAL(V$):
   JS=JS+1:
   JR=4
9720 IF JS>6 THEN JS=2
9730 GOTO 9010
```

Listing 7-2. Plotting program for forecasting.

```
1 REM PLOT PROGRAM FOR FORECASTING
2 REM VERSION 1.1 BY LES
3 REM FOR TRS-80 MODEL 100
4 REM READS DATA ENTERED USING DATA
5 REM ENTRY PROGRAM
9 DEFSNG A-Z
10 CLS
20 DIM X(100),Y(100)
21 INPUT"FILE NAME ";F$:
   PRINT"PRESS C FOR CASSETTE R FOR RAM:
   ";:
   Y$=INPUT$(1)
22 IF Y$="R"THENF$="RAM:
   "+F$+".DO" ELSEF$="CAS:
   "+F$
24 OPENF$FORINPUTAS1:
   INPUT#1,T$,R$,X$,Y$,NY:
   CLS
30 CLS
```

```
31 PRINT T$:
   PRINTR$:
   PRINTX$:
   PRINTY$:
   PRINT"IS THIS THE FILE YOU WANT Y OR N"
32 Y$=INKEY$:
   IFY$=""THEN32ELSE IFY$="N"THENCLOSE:
    END
33 PRINT"IS THIS TIME SERIES Y OR N";:
   Y$=INPUT$(1):
   IFY$="Y"THEN 34 ELSE 36
34 FORJ=1TONY:
   INPUT#1,X(J),Y(J):
   X(J)=J:
   NEXTJ:
   CLOSE
35 GOTO 40
36 FORJ=1TONY:
   INPUT#1,X(J),Y(J):
   NEXTJ:
   CLOSE
40 XI=1E9:
   XM=-1E9:
   YM=XM:
   YI=XI
50 FORJ=0TONY-1
60 IFX(J)>XMTHENXM=X(J)
70 IFX(J)<XITHENXI=X(J)
80 IFY(J)>YMTHENYM=Y(J)
90 IF Y(J)<YITHENYI=Y(J)
100 NEXTJ
115 IF XM=XITHENXM=2*XI:
   XI=.5*XI
116 IF YI=YMTHENYM=2*YI:
   YI=.5*YI
120 DX=XM-XI:
   DY=YM-YI:
   SX=180/DX:
   SY=55/DY
130 PRINTDX,SX,DY,SY
140 REM scale
145 CLS
150 PRINT@284,XI;:
   PRINT@313,XM;
240 PRINT@240,;:
   PRINTUSING"###.##";YI;:
   PRINT@0,;:
   PRINTUSING"###.##";YM;
245 LINE(34,56)-(234,56)
246 LINE (34,56)-(34,0)
247 FOR J=0TONY-1:
   YP=Y(J)-YI:
   YP=YP*SY:
   YP=56-YP:
   XP=X(J)-XI:
   XP=34+XP*SX
248 IF J=0THENX1=XP:
   Y1=YP:
   GOTO252
249 LINE(X1,Y1)-(XP,YP)
250 X1=XP:
   Y1=YP
252 NEXTJ
260 X=33:
   LINE(X,55)-(X,56):
   FORJ=1TONY-1:
   X=X+SX:
   LINE(X,55)-(X,57):
   NEXTJ
270 PRINT@80,C$;
280 Y$=INKEY$:
   IFY$=""THEN280
290 GOTO 30
```

Listing 7-3. Linear regression program LINREG.BA.

```
10 REM LEAST SQUARES CURVE FIT
20 REM VERSION 1.1 NOV 1983
30 REM FOR TRS-80 MODEL 100 BY LES
40 REM GIVES Y = A + BX
50 REM FOR TAB BOOKS
60 REM REQUIRES DATA FILE CREATED BY
70 REM DATA ENTRY PROGRAM
90 REM *****************************
100 CLS:
   CLEAR:
   PRINT STRING$(39,"=")
105 DIM X(100),Y(100)
110 PRINT:
   PRINT "LINEAR LEAST SQUARES
```

```
120 PRINT:
   PRINTSTRING$(39,"=")
130 INPUT"ENTER NAME OF DATA FILE ";DF$
140 PRINT "PRESS C FOR CASSETE R FOR RAM";:
   Y$=INPUT$(1):
   IFINSTR("CR",Y$)=0THEN140
150 IF Y$="R" THEN DF$=DF$+".DO" ELSE DF$="CA
      :
   "+DF$
175 OPENDF$FORINPUTAS1:
   INPUT#1,R$,T$,X$,Y$,N:
   CLS:,
   PRINT"REMARKS ";R$:
   PRINT"PERIOD COVERED ";T$:
   PRINT "NUMBER OF POINTS ";N:
   PRINT"X VARIABLE ";X$:
   PRINT"Y VARIABLE ";Y$:

176 PRINT"IS THIS THE FILE YOU WANT?":
   Z$=INPUT$(1):
   IFZ$="Y"ORZ$="y"THEN177ELSE CLOSE:
   END
177 FORJ=1TON:
   INPUT#1,X(J),Y(J):
   NEXTJ:
   CLOSE
300 REM NOW DO CALCULATIONS
310 A1=0
320 A2=0
330 B1=0
340 B2=0
345 SY=0
350 FOR J = 1 TO N
360    A1=A1+X(J) :
    A2=A2+X(J)*X(J)
370    B1=B1+Y(J):
   B2=B2+Y(J)*X(J)
375    SY=SY+Y(J)*Y(J)
380 NEXT J
400 B =(N*B2-A1*B1)/(N*A2-A1*A1)
410 A = (B1-B*A1)/N
430 SU=0
440 FORJ=1TON-1
450  S1=Y(J)-A-B*X(J)
460  SU = SU+S1*S1
```

```
470 NEXTJ
480 SD=SQR(SUM/(N-2))
490 CLS:
   PRINT"RESULTS"
500 PRINTUSING"THE EQN IS Y=####.### + ####.##
     #X";A,B
510 PRINTUSING"STANDARD DEVIATION ###.####";SD

520 R= (B2-A1*B1/N)/(SQR((A2-A1*A1/N)*(SY-(B1*
     B1)/N)))
530 PRINTUSING"R SQUARED = ####.####";R*R
540 PRINT:
   PRINT"DO YOU WANT TO FORECAST?"
550 Y$ = INKEY$:
   IFY$=""THEN550
560 IF Y$="N" OR Y$="n" THEN END
570 CLS
580 PRINT"FORECAST VALUE OF Y FOR GIVEN X."
590 INPUT "ENTER VALUE OF X ";X
600 PRINTUSING" FOR X =####.#### Y = #####,#.#
     ##";X,A+B*X
610 GOTO 540
```

Listing 7-4. Program for exponential smoothing.

```
10 REM EXPONENTIAL SMOOTHING FORECAST
20 REM WITH TRACKING SIGNAL
25 REM IF TRACKING SIGNAL >0.7
26 REM SYSTEM IS OUT OF CONTROL
30 REM VERSION 1.0 11/83
40 REM BY L.E.SPARKS
50 REM USES DATA FILE IN RAM
60 REM **************************
65 F$="#####.##":
   S$="#####.##":
   T$="#.###"
70 CLS:
   PRINT"EXPONENTIAL FORECASTING"
75 PRINT
80 PRINT"ENTER DATA FILE NAME";:
   INPUTA$
82 A$=A$+".DO"
85 PRINT"IF 1ST TIME PRESS1 ELSE ANYOTHER K
     EY"
86 Y$=INPUT$(1):
   IFY$="1" THEN110ELSE90
```

```
90 OPENA$FORINPUT AS 1
100 INPUT#1,A,E1,F,M1:
   CLOSE
105 CLS:
   GOTO130
110 INPUT"ENTER SMOOTHING CONSTANT ";A
111 M1=12.5:
   E1=1.25
120 INPUT "ENTER 1ST FORECAST ";F
130 INPUT "ENTER ACTUAL VALUE FOR PERIOD";Y

135 CLS
140 F2=A*Y+(1-A)*F
150 E=Y-F:
   E2=A*E+(1-A)*E1:
   M=A*ABS(E)+(1-A)*M1:
   S=1.25*M:
   T=E2/M
160 PRINT "FORECAST FOR NEXT PERIOD :
   "
170 PRINTUSINGF$;F2
180 PRINT "ESTIMATED STD DEVIATION "
190 PRINTUSINGS$;S
200 PRINT "TRACKING SIGNAL ";:
   PRINTUSINGT$;T
210 IF ABS(T)>.7THEN PRINT"warning out of c
     ontrol"
240 OPENA$FOROUTPUTAS1
250 PRINT#1,A,E2,F2,M:
   CLOSE
```

Listing 7-5. Adaptive filtering program ADAPT.BA .

```
10 CLEAR 100
12 TI$=TIME$
160 DEFINT I,J,L,N,M
170 DIM X(100),Y(100),W(50),YN(100),F(100),E(
     100)
171 CLS:
   PRINT"ADAPTIVE FILTERING PROGRAM":
   PRINT"FOR MODEL 100 BY LES."
172 INPUT"ENTER FILE NAME :
   ";F$
173 PRINT"<C>ASSETE OR<R>AM FILE":
   PRINT"PRESS C OR R"
```

```
174 Z$=INPUT$(1):
   J=INSTR("CcRr",Z$):
   IFJ=0THEN174ELSEIFJ<3THENF$="CAS:
   "+F$ELSEF$="RAM:
   "+F$+".DO"
175 OPENF$FORINPUTAS1:
   INPUT#1,R$,T$,X$,Y$,N:
   CLS:
   PRINT"REMARKS ";R$:
   PRINT"PERIOD COVERED ";T$:
   PRINT "NUMBER OF POINTS ";N:
   PRINT"X VARIABLE ";X$:
   PRINT"Y VARIABLE ";Y$:
176 PRINT"IS THIS THE FILE YOU WANT?":
   Z$=INPUT$(1):
   IFZ$="Y"ORZ$="y"THEN177ELSE CLOSE:
   END
177 FORJ=1TON:
   INPUT#1,X(J),Y(J):
   NEXTJ:
   CLOSE
300 CLS
320 GOSUB 1190
340 INPUT "ENTER # OF PERIODS ";P
350 K=1/P
370 FOR J=1TON:
    W(J)=J/P:
    NEXTJ
380 L=200
410 CLS
420 PRINT "WORKING"
430 Z=1E37
440 FOR L1=1TOL
450 S=0
460 H1=0
470  FOR I=P+1TON
480   F=0
490   FORJ1=1TOP
500     F=F+W(J1)*YN(I-J1)
510     H1=H1+YN(I-J1)*YN(I-J1)
520   NEXT J1
530 H1=SQR(H1)
540 E(I)=YN(I)-F
550 FORJ=1TOP
560  W(J)=W(J)+2*K*E(I)/H1*YN(I-J)/H1
570 NEXT J
```

```
580 S=S+E(I)*E(I)
590 NEXTI
650 IF S+.00001>ZTHEN680
655 PRINTUSING "ITERATION ## OF #### ";L1,L
660 Z=S
670 NEXTL1
680 CLS
685 T2$=TIME$
690 INPUT "NUMBER OF FORECASTS ";M
700 CLS
710 FORI=N+1TON+M
720 F=0
730 FORJ=1TOP
740  F=F+W(J)*YN(I-J)
750 NEXT J
760 F(I)=F*HI
770 YN(I)=F
780 NEXTI
790 CLS
810 PRINT "RESULTS
820 FOR J=N+1 TO N+M
840 PRINT J,F(J)
850 NEXT J
1180 END
1190 REM
1200 HI=Y(1)
1210 FOR I=2TON
1220   IFY(I)>HITHENHI=Y(I)
1230 NEXT I
1240 FOR I=1 TO N
1250   YN(I)=Y(I)/HI
1260 NEXT I
1270 RETURN
```

Appendix A

Translating for Other Computers

The programs are written in Microsoft BASIC and can be modified to run on other computers that use BASIC. The main difficulties in translating the programs to other dialects of BASIC are in the data entry and the data display modules. These are the two modules that use hardware specific features of Microsoft BASIC. This appendix will discuss three ways of dealing with these modules.

The easiest way to modify the programs for other computers is to replace the data entry module with a module based on INPUT commands. This eliminates the need to reformat the screen for other computers. This method also eliminates the data entry forms which are one of the best features of the programs. An example of the replacement module for a loan program is given below.

```
9000 REM REPLACEMENT DATA ENTRY MODULE

9010 REM EXAMPLE SHOWING USE OF INPUT COMMANDS

9020 REM EXAMPLE FOR LOAN PROGRAM

9030 PRINT "CLEAR SCREEN":REM USE CLS FOR IBM AND TRS-80 HOME FOR APPLE

9040 REM PRINT CHR$(27)CHR$(42) FOR MANY TERMINALS

9050 PRINT "DATA ENTRY FOR LOAN PROGRAM. ENTER DATA ASKED FOR."

9060 PRINT "THE PRESENT VALUE IS SHOWN IN () "
```

```
9070 FOR J = 1 TO 4

9080      PRINT X$(J);" (";X(J);")";

9090      INPUT X(J)

9100 NEXT J

9110 PRINT "CLEAR SCREEN":REM USE THE APPROPRIATE COMMAND

9120 PRINT "TO MAKE CORRECTIONS ENTER THE NUMBER CORRESPONDING TO THE
     INCORRECT ITEM. ENTER Y TO CONTINUE"

9130 FOR J = 1 TO 4

9140    PRINT X$(J); "  ";X(J)

9150 NEXT J

9160 INPUT A$

9170 IF A$ = "Y" THEN RETURN

9180 J = VAL(A$) : IF J = 0 OR J > 4 THEN 9150

9190 PRINT "ENTER ";X$(J);

9200 INPUT X(J)

9210 GOTO 9110

9990 REM INITIALIZE ARRAYS FOR DATA ENTRY

9991 DATA AMOUNT OF LOAN, INTEREST RATE, NUMBER OF PERIODS/YR, NUMBER OF
     YEARS, AMOUNT OF PAYMENT

9992 FOR J = 1 TO 5

9994    READ X$(J)

9995 NEXT J

9996 RETURN
```

The display program would look something like this:

```
2000 REM EXAMPLE OF DISPLAY PROGRAM

2010 PRINT "CLEAR SCREEN

2020 PRINT "RESULTS OF LOAN PROGRAM "

2030 PRINT

2040 FOR J = 1 TO 5

2060    PRINT X$(J);" ";X(J)

2070 NEXT J

2080 PRINT "PRESS ENTER TO CONTINUE ";

2090 INPUT A$

2100 RETURN
```

The structure of the main line program would look like this.

```
10 REM STRUCTURE OF LOAN PROGRAM

20 GOSUB 9990 :REM INITIALIZE

30 GOSUB 9000 :REM GET DATA

40 GOSUB 100  :REM DO CALCULATIONS

50 GOSUB 2000

60 END
```

This method of data entry is shorter than the data entry module. Note that corrections can be made only after all the data are entered. Also note that the display module displays all the input data as well as the calculated answer.

The modification necessary to use the data entry and display modules for the TRS-80 Models I, III, and 4 are simple. The commands these computers use to address the display are the same as those used by the Model 100. The only difference is the size of the display. To use the programs on these three computers modify the PRINT@statements by substituting 64 for 40. For example, the Model 100 statement

```
9010 PRINT @JS*40, X$(1)
```

corresponds to the TRS-80 Model I, III, and 4 statement

```
9010 PRINT @JS*64,X$(J)
```

The other change necessary is to change the PRINT @ 280,“ ”; statements to

```
PRINT @ 896,“”
```

The final changes deal with the commands to move the cursor. The arrow keys in the Model 100 have different ASCII values than the arrow keys on the Models, I, III, and 4. Table A-1 below shows the corresponding values.

Because the shift down arrow can not be used on the Models I, III, and 4, some other key must be used to change pages. I suggest the > key be used.

The changes necessary to use the arrow keys in the Models I, III, and 4 are made in the control string AZ$. Where the Model 100 listing shows CHR$(28), use CHR$(9), and so on.

Some versions of BASIC provide screen addressing with a LOCATE command. Usually the locate command requires that you tell the computer the row and line that you are to start printing at. If your version of BASIC has such a command, replace the PRINT@commands with a LOCATE x,y: PRINT sequence. You will have to experiment to find the proper combination of x and y to get the display to your liking. You will also have to change the control string AZ$ to account for the different values of the arrow keys.

Table A-1. Comparison of Cursor Character Codes.

Key	Model 100	Models I, III, and 4
Right arrow	28	9
Shift right arrow	6	25
Left arrow	29	8
Shift left arrow	1	24
Up arrow	30	91
Shift up arrow	20	27
Down arrow	31	10
Shift down arrow	2	cannot use

Note that some versions of BASIC replace the PRINT @ command with PRINT x,y instead of the LOCATE x,y command.

The plot program will require extensive modification to run on a computer other than the Model 100. Detailed instructions for this are outside the scope of this appendix.

Appendix B

Additional Reading

The references listed below provide additional discussion of the topics covered in this book.

Gibbs, John and McGee, Robert W. *Financial Decision Making in Business.* Englewood Cliffs, NJ: Prentice-Hall, 1980.

Glahe, Fred R. *Macroeconomics, Theory and Policy.* New York: Harcourt Brace Jovanovich, Inc., 1977.

Haley, Charles W., and Schall, Lawrence D. *The Theory of Financial Decisions.* New York: McGraw-Hill, 1979.

Humphrey, Thomas J. *Basic Programming for the Financial Executive.* New York: John Wiley and Sons, 1978.

Nerlove, Marc; Grether, David M.; and Carvalho, Jose L. *Analysis of Economic Time Series:* New York: Academic Press, 1979.

Pindyck, R. S., and Rubinfeld, D. L. *Econometric Models and Economics Forecasts.* New York: McGraw-Hill, 1976.

Pring, Martin J. *How to Forecast Interest Rates.* New York: McGraw-Hill, 1981.

Rachlin, Robert. *Return on Investment.* Englewood Cliffs, NJ: Prentice Hall, 1976.

Rosefsky, Robert S. *Personal Finance and Money Management.* New York: John Wiley and Sons, 1978.

Schmeltz, L. R. *Playing the Stock and Bond Market with Your Personal Computer.* Blue Ridge Summit, PA: TAB Books, 1981.

Seitz, Neil. *Financial Analysis.* Reston, VA: Reston Publishing, 1979.

Sparks, L.E. *Investment Analysis with Your Microcomputer.* Blue Ridge Summit, PA: TAB Books, 1983.

Wheelwright, Steven C., and Makridakis, Spyros. *Forecasting Methods for Management, 3rd Ed.* New York: John Wiley and Sons, 1980.

Index

Financial Decision Making with Your TRS-80™ Model 100, including 18 Programs

If you are intrigued with the possibilities of the programs included in *Financial Decision Making with Your TRS-80™ Model 100, including 18 Programs* (TAB Book No. 1799), you should definitely consider having the ready-to-run tape containing the software applications. This software is guaranteed free of manufacturer's defects. (If you have any problems, return the tape within 30 days, and we'll send you a new one.) Not only will you save the time and effort of typing the programs, the tape eliminates the possibility of errors that can prevent the programs from functioning. Interested?

Available on tape for Radio Shack TRS-80™ Model 100 with 8K or 24K at $19.95 plus $1.00 each shipping and handling.

I'm interested in the ready-to-run tape for *Financial Decision Making with Your TRS-80™ Model 100, including 18 Programs*. Send me:

______ tape for TRS-80™ Model 100, 8K or 24K (6054S)

______ TAB BOOKS catalog

______ Check/Money Order enclosed for $19.95 plus $1.00 shipping and handling for each tape ordered.

______ VISA ______ MasterCard

Account No. ________________________ Expires __________

Name ____

Address ____

City ________________________ State __________ Zip ________

Signature ____

Mail To: **TAB BOOKS Inc.**
P.O. Box 40
Blue Ridge Summit, PA 17214

(Pa. add 6% sales tax. Orders outside U.S. must be prepaid with international money orders in U.S. dollars.)

TAB 1799

Financial Decision Making with Your TRS-80™ Model 100, including 18 Programs

by Leslie Sparks

Entrepreneur . . . executive . . . broker . . . banker . . . businessman . . . homeowner . . . *anyone* who wants to make his money work harder and who wants to have better control over his financial future will appreciate this extremely well-written and eminently practical sourcebook!

It's an easy-to-follow guide to transforming your TRS-80 Model 100 into a powerful and reliable assistant for financial planning and decision making. You'll find complete listings for a series of programs that make even the most complicated calculations and number manipulations a snap. And you'll find all the explanations you need on how and why these programs can help you handle financial problems that might have required a private accountant in pre-micro days!

This is where you'll find straightforward, step-by-step information on basic program structure and design, on the use of the data entry module, and other programming tips. Then, you'll get a full description of the fundamentals of time and money, present value and financial decision making, breakeven analysis, ratio analysis, forecasting, and graphing.

The programs themselves–developed by an author who is well-versed in both financial planning and investment analysis as well as computer programming—are fully functional and designed to give a clean, concise screen layout. This latter feature is particularly important for the Model 100 with its minimum screen display. Plus, the author has included easy-to-use directions for converting these Model 100 programs to the BASIC used on other microcomputers.

Whether your financial needs are confined to calculating savings and loan interest rates . . . or you're embarking on a major financial investment program, this exceptionally complete guidebook will make it easier for you to control your money and to get the most from your hard-earned dollars!

Leslie Sparks has a Ph.D. in Engineering from the University of Washington and is involved in utility research and development. He has written numerous articles for microcomputing magazines and technical journals and is the author of TAB's *Investment Analysis with Your Microcomputer.*

TAB **TAB BOOKS Inc.**
Blue Ridge Summit, Pa. 17214

FPT >> 9.95

ISBN 0-8306-1799-X

PRICES HIGHER IN CANADA

965-0884